



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

MODEL REDUCTIONS FOR QUEUEING AND  
AGENT-BASED SYSTEMS  
WITH APPLICATIONS IN COMMUNICATION  
NETWORKS

vom Fachbereich Elektrotechnik und Informationstechnik der  
TECHNISCHE UNIVERSITÄT DARMSTADT

zur Erlangung des Grades  
Doktor rerum naturalium (Dr. rer. nat.)  
Dissertation

von

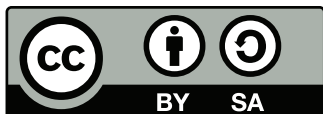
WASIUR RAHMAN KHUDA BUKHSH, M.STAT.

Erstgutachter: Prof. Dr. techn. Heinz Köppl  
Zweitgutachter: Prof. Dr. -Ing. Ralf Steinmetz  
Drittgutachter: Prof. Dr. rer. nat. Frank Aurzada

Darmstadt 2018

The work of Wasiur Rahman Khuda Bukhsh was supported by the German Research Foundation (DFG) in the Collaborative Research Centre (SFB) 1053 "MAKI - Multi-Mechanism-Adaptation for the Future Internet" at the Technische Universität Darmstadt, Germany.

Khuda Bukhsh, Wasiur Rahman : *Model reductions for queueing and agent-based systems with applications in communication networks*  
Darmstadt, Technische Universität Darmstadt  
Jahr der Veröffentlichung der Dissertation auf TUpriints: 2018  
Tag der mündlichen Prüfung: 28.06.2018



Veröffentlicht unter CC BY-SA 4.0 International  
<https://creativecommons.org/licenses/by-sa/4.0/>

Dedicated to my mother.



## ABSTRACT

---

The dissertation studies two distinct classes of models from applied probability literature and attempts to answer questions that are of asymptotic nature. The answers to those questions are obtained by means of various probability approximations. The two classes of models in question belong to the mathematical branches of queueing theory and Markovian Agent-based Models (MABMs). The unlikely marriage of these two branches of probability theory in this dissertation can be ascribed to a particular communication networking scenario, the mathematical modelling of which has been the main motivation behind this work. The networking scenario consists of two central problems - the uploading problem and the content distribution problem.

In the context of Internet of Things (IoT), collaborative uploading describes a type of crowdsourcing scenario in networked environments where a device utilises multiple paths to upload content to a centralised processing entity such as a cloud service. The *uploading problem* is an umbrella term for research problems arising from such a collaborative uploading scenario and encompasses questions such as how long it will take for a data chunk to be transported, how many paths we should choose (scheduling), how to split a data chunk optimally (provisioning). Modelling the uploading problem as a scheduling task in a Fork-Join (FJ) system, a parallel queueing system with output synchronisation, we develop the notions of optimal stochastic scheduling, and provisioning. Since an exact analysis of FJs system is infeasible under general settings, the objectives of designing optimal stochastic schedules and provisions are achieved by approximating probabilities of rare events (*e.g.*, long waiting times) with exponential estimates. This is accomplished by making use of martingale techniques and establishing a Large Deviations Principle (LDP) for steady-state waiting times. In order to incorporate possible burstiness or phase-type behaviour, the effects of changing environment are modelled using a Markov-additive process. The resultant theoretical insights are finally used to design optimal collaborative uploading strategies.

In addition to general FJ systems, two special queueing systems are analysed using random time change representation for Markov processes in this dissertation. Unlike FJ systems, these two special queueing systems do not impose an inherent output synchronisation. The first of these two special cases is a parallel queueing system with finite buffers. Preliminary ideas on characterising the total loss process and optimal probabilistic scheduling are presented. As an application to large heterogeneous clusters of parallel servers, we also present a scaling limit as the number of servers increases to infinity using the semigroup operator approach to Markov process convergence. The second queueing system considers a special case of the uploading problem where the paths can transport *only* one chunk of data at a time. We use multi-scaling techniques from probability theory to derive Quasi-Steady State Approximations (QSSAs) for such a queueing system. The QSSAs are particularly useful when the number of data chunks to transport is much larger compared to the number of available paths.

The second leg of the networking scenario, the *distribution problem*, concerns distribution of content to a number of end-users. We specifically focus on the large-scale prob-

lem when the number of end-users is large. In order to understand the dynamics of the distribution problem better, we model it as an MABM. Three different approximations for MABMs are presented in this dissertation. First, a Functional Central Limit Theorem (FCLT) for key population counts are proved for an Information-Dissemination (ID) process on configuration model random graphs. An ID process is mathematically equivalent to a stochastic compartmental Susceptible-Infected (SI) epidemic process. Second, we devise a state-aggregation procedure based on a local notion of symmetry (automorphism) of the underlying graph for general MABMs ensuring approximate lumpability. Third, as an application, primitive chunk selection strategies for Peer-to-Peer (P2P) live streaming systems, such as the Latest Deadline First (LDF) and the Earliest Deadline First (EDF), are analysed using mean-field theory and an improved mixed strategy, called SCHEDMIX, is proposed.

As the mathematical models are developed in response to the questions arising from the communication networking scenario, special emphasis has been put on exploring how the resultant approximation tools can also be applied to problems in epidemiology, systems and synthetic biology, statistical physics, and other branches of science.

## ZUSAMMENFASSUNG

---

Die Dissertation untersucht zwei verschiedene Klassen von Modellen für Warteschlangen- und Agenten-basierte Systeme aus der Literatur der angewandten Wahrscheinlichkeitstheorie und versucht Fragen zu beantworten, die weitgehend von asymptotischer Natur sind. Die betrachtete Verflechtung dieser beiden unterschiedlichen Zweige der Wahrscheinlichkeitstheorie ist auf ein bestimmtes Kommunikationsnetzwerkszenario zurückzuführen, dessen mathematische Modellierung die Hauptmotivation dieser Arbeit ist. Das Netzwerkszenario besteht aus zwei zentralen Problemen - dem Hochladeproblem und dem Inhaltsverteilungsproblem.

Im Kontext des Internet der Dinge (Internet of Things) beschreibt kollaboratives Hochladen eine Art von Crowdsourcingszenario in vernetzten Umgebungen, in denen ein Gerät mehrere Pfade zum Hochladen von Inhalten in eine zentrale Verarbeitungsentität wie einen Cloud-Service verwendet. Das *Hochladeproblem* ist ein Sammelbegriff für Forschungsprobleme, die sich aus einem solchen Szenario ergeben, und umfasst unter anderem die Fragen 1) Wie lange dauert es, um einen Datenblock zu transportieren? 2) Wie viele Pfade sollten ausgewählt werden (*Scheduling*) um einen Datenblock optimal aufzuteilen (*Provisionierung*)? Indem wir das Hochladeproblem als Scheduling-Task in einem Fork-Join (FJ)-System, nämlich einem parallelen Warteschlangensystem mit Ausgangssynchronisation, modellieren, entwickeln wir die Begriffe des optimalen stochastischen Scheduling und der Provisionierung. Da eine exakte Analyse des FJ-Systems unter allgemeinen Annahmen über die Ankünfte und den bereitgestellten Dienst sich bislang als hoch diffizil herausgestellt hat, werden die Ziele, das optimale stochastische Scheduling und die Provisionierung zu entwerfen, durch Näherungen von Warscheinlichkeiten seltener Ereignisse (z. B. lange Wartezeiten) mit exponentiellen Schätzungen erreicht. Dies wird durch die Nutzung von Martingaltechniken und die Einführung eines Large Deviations Principle (LDP) für stationäre Wartezeiten erzielt. Um ein mögliches Burst- oder Phasenverhalten zu berücksichtigen, werden die Auswirkungen sich ändernder Umgebungsbedingungen mit Hilfe eines Markov-additiven Prozesses modelliert. Die daraus resultierenden theoretischen Erkenntnisse werden schlussendlich verwendet, um optimale Strategien für kollaboratives Hochladen zu entwickeln.

Zusätzlich zu allgemeinen FJ-Systemen werden zwei spezielle Warteschlangensysteme unter Verwendung einer zufälligen Zeitänderungsdarstellung für Markov-Prozesse in dieser Dissertation analysiert. Im Gegensatz zu FJ-Systemen erzwingen diese beiden speziellen Warteschlangensysteme keine inhärente Ausgangssynchronisation. Der erste dieser beiden Spezialfälle ist ein paralleles Warteschlangensystem mit endlichen Puffern. Hier werden Methoden zur Charakterisierung des Verlustprozesses und der optimalen probabilistischen Planung in einem solchen endlichen Puffer-Warteschlangensystem vorgestellt. Als Anwendung für große heterogene Cluster von parallelen Servern wird eine Skalierungsgrenze für eine ansteigende Anzahl der Server vorgestellt. Das zweite Warteschlangensystem berücksichtigt einen Spezialfall des Hochladeproblems, bei dem die Pfade keine Pufferungsmöglichkeiten haben. Wir verwenden Multi-Skalierungstechniken aus der Wahrscheinlichkeitstheorie, um Quasi-Steady State Approximations (QSSAs) für ein solches Warteschlangensystem abzuleiten. Die QSSAs sind besonders nützlich, wenn

die Anzahl der zu transportierenden Datenblöcke viel größer ist als die Anzahl der verfügbaren Pfade.

Der zweite Teil des Netzwerkszenarios, das *Inhaltsverteilungsproblem*, betrifft die Verteilung von Inhalten in vernetzten Umgebungen zu einer Vielzahl von Endnutzern. Hier gehen wir hauptsächlich auf die Problemstellung für eine große Anzahl von Endnutzern ein. Um die Dynamik des Verteilungsproblems besser zu verstehen, modellieren wir es als ein Markovian Agent-based Model (MABM). In dieser Dissertation werden drei verschiedene Approximationen für MABMs vorgestellt. Zuerst wird ein Functional Central Limit Theorem (FCLT) für wichtige Populationsvariablen für einen Information-Dissemination (ID)-Prozess auf zufälligen Graphen des Konfigurationsmodells bewiesen. Der Information-Dissemination (ID)-Prozess ist mathematisch äquivalent zu einem stochastischen Compartmental Susceptible-Infected (SI) epidemischen Prozess. Zweitens wird ein Zustands-Aggregationsverfahren basierend auf den lokalen Symmetrien des zugrunde liegenden Graphen für generelle MABMs entwickelt, um die approximative *Lumpability* sicherzustellen. Drittens werden als eine Anwendung primitive Paket-Auswahlstrategien für Peer-to-Peer (P2P) Live-Streaming-Systeme, wie zum Beispiel die Latest Deadline First (LDF) und die Earliest Deadline First (EDF), unter Verwendung der Mean-Field-Theorie analysiert und eine verbesserte gemischte Strategie, das sogenannte SCHEDMIX, wird vorgeschlagen.

Während unsere mathematischen Modelle Fragen adressieren, die sich aus dem Kommunikationsnetzwerkszenario ergeben, untersuchen wir die Tragweite der resultierenden Approximationswerkzeuge bei zusätzlicher Anwendung auf Probleme der Epidemiologie, der Systembiologie und synthetischen Biologie, der statistischen Physik und anderer Gebieten der Wissenschaft.



## ACKNOWLEDGEMENTS

---

First and foremost, my sincerest thanks to Prof. Heinz Koepl for patiently guiding me through all the stages of doctoral studies, for giving me the time and freedom to pursue my research interests, for his relentless encouragement, and for the wonderful opportunity that this has been. I would also like to tender my sincere thanks to Prof. Ralf Steinmetz and Prof. Frank Aurzada for being my co-referees, and Prof. Abdelhak M. Zoubir, Prof. Florian Steinke and Prof. Rolf Jakoby for serving on my PhD committee.

All the past and present members of the BCS lab have been wonderful. Special thanks to Markus Baier and Christine Cramer for always helping with technical and administrative matters. Bastian Alt, Leo Bronstein, Jascha Diemer, Maleen Hanst, Nikita Kruk, Francois-Xavier Lehr, Dominik Linzner, Tim Prangemeier, Adrian Šošić, Sikun Yang, Christian Wildner, Mark Sinzger, Hameer Abbasi, Ranjani Krishnan, Volodymir Volchenko, Tabea Treppmann, Sara Al-Sayed, Nurgazy Sulaimanov, Derya Altintan - you all are special, many thanks for being so affable.

I gratefully acknowledge all my collaborators, each of whom introduced me to new perspectives and taught me different ways of doing science. Special thanks to Amr Rizk, Greg Rempala, Hye-Won Kang, Yann Disser, Casper Woroszylo, Alexander Frömmgen, and Julius Rückert. I also thank the Mathematical Biosciences Institute (MBI) for the warm hospitality I received during my visits there.

I am deeply indebted to all my friends, without whom life would be so much more monotonous and at times, difficult. Infinitely many thanks to Nabamita Saha, Namrata Chakraborty, Aurindam Dhar, Bishakh Bhattacharya, Kaunteya Guha, Soumik Sao, Arnab Banerjee, Rajeev Biswas, Sounak Kar, Arindam Fadikar, Tanmoy Maity, Pushkar Singh, Sunil Kumar, Sanchari Bhattacharya, Kriti Sharma, Goutam Das, Indrajit Jana, Saswata Adhikary. I must thank Ranjan Dutta, who first encouraged me to study statistics! Many many thanks to my Persian friends' gang in Germany - Mehran Sarabchian, Mahsa Hajiani, Habib Pouriaeyali, Saeed Ehteshamifar. I wish I did not have to part.

Respectful pranam to all my teachers for being not just great teachers but also great human beings. Many thanks to Subir Bhandari, Bikas Sinha, Pradipta Maji, Prasanta Kumar Giri, Nanda Kishore De, Dilip Kumar Sahoo, Parthosarathi Chakraborti, Tulsi-das Mukhopadhyay, Subhadeep Banerjee, Palas Pal, Satyaki Pal, Sudipto Chakraborty, Subrata Biswas, Krishnendu Bandyopadhyay, and Sushil Kumar Ghosh. I wish I could ever be as selfless, as righteous, as kind and as loving as the two monks Swami Suparnananda (Satya da) and Swami Jnanalokananda, both of whom had a tremendous influence on my life.

Finally, I dedicate this dissertation to my mother, Saira Banu, my greatest source of inspiration. It would not have been possible without her countless sacrifices, and inexhaustible love, support and encouragement.



# CONTENTS

---

1	INTRODUCTION	1
1.1	Motivation: a communication networking scenario . . . . .	1
1.2	The uploading problem from a queueing perspective . . . . .	3
1.3	The distribution problem as a Markovian Agent-based model . . . . .	7
1.4	Organisation of the thesis . . . . .	8
1.5	Publications . . . . .	9
2	PRELIMINARIES	11
2.1	Notational conventions . . . . .	11
2.2	Fork-Join queues . . . . .	11
2.3	Chemical reaction networks and queueing systems . . . . .	13
2.4	Lumpability . . . . .	17
2.5	Markovian agent-based models . . . . .	19
3	STOCHASTIC SCHEDULING IN FORK-JOIN QUEUEING SYSTEMS	21
3.1	Heterogeneous Fork-Join queueing systems . . . . .	21
3.2	Scheduling tasks in heterogeneous FJ systems . . . . .	23
3.3	Scheduling under application specific scaling . . . . .	24
4	PROVISIONING IN FORK-JOIN QUEUEING SYSTEMS	33
4.1	Markov-additive process formulation . . . . .	33
4.2	Blocking systems . . . . .	43
4.3	Fork-Join System with non-renewal input . . . . .	45
4.4	Parallel Systems with Dependent Servers . . . . .	46
4.5	Markov Modulated Arrivals and Service . . . . .	49
4.6	Further extensions . . . . .	50
5	COLLABORATIVE UPLOADING	53
5.1	Modelling approach . . . . .	53
5.2	Intermittent collaborative uploading . . . . .	54
5.3	Stream Uploading . . . . .	59
6	SCHEDULING IN FINITE-BUFFER QUEUEING SYSTEMS	63
6.1	Model . . . . .	63
6.2	Optimal scheduling in $N$ -server queues with finite buffers . . . . .	66
6.3	Queueing systems with exogenous modulation . . . . .	67
6.4	A scaling limit: an application to clusters of shared servers . . . . .	69
7	QUASI-STEADY STATE APPROXIMATIONS	77
7.1	Why QSSA? . . . . .	77
7.2	QSSAs for deterministic Michaelis-Menten kinetics . . . . .	79
7.3	Multi-scale stochastic Michaelis-Menten kinetics . . . . .	81
7.4	Standard quasi-steady-state approximation . . . . .	85
7.5	Total quasi-steady-state approximation . . . . .	88
7.6	Reverse quasi-steady-state approximation . . . . .	92
7.7	Discussion . . . . .	96
8	A FUNCTIONAL CENTRAL LIMIT THEOREM	99
8.1	Model . . . . .	99

8.2	The law of large numbers . . . . .	101
8.3	Functional central limit theorem . . . . .	103
8.4	Applications . . . . .	109
8.5	Discussion . . . . .	117
9	APPROXIMATE LUMPABILITY . . . . .	119
9.1	Markovian Agent-based Model . . . . .	119
9.2	Automorphism-based lumping of an MABM . . . . .	122
9.3	Lumping states using local symmetry . . . . .	125
9.4	Graph fibrations . . . . .	128
9.5	Approximation error . . . . .	129
9.6	Discussions . . . . .	136
10	P2P LIVE STREAMING . . . . .	139
10.1	Model . . . . .	139
10.2	Mean-field theoretic analysis . . . . .	142
10.3	Simulation results . . . . .	151
10.4	Discussion . . . . .	152
11	CONCLUDING REMARKS . . . . .	155
11.1	Summary of contributions . . . . .	155
11.2	Future directions . . . . .	157

## Appendices

A	STOCHASTIC SCHEDULING . . . . .	161
A.1	Main proofs . . . . .	161
A.2	Statistical results . . . . .	164
A.3	Service time scaling . . . . .	165
A.4	Evaluation of deterministic and stochastic strategies . . . . .	169
B	PROVISIONING . . . . .	171
B.1	Large deviations principle . . . . .	171
B.2	Further derivations . . . . .	172
C	COLLABORATIVE UPLOADING . . . . .	175
C.1	Statistical results . . . . .	175
C.2	Additional derivations . . . . .	177
C.3	Rigid allocation . . . . .	182
D	FINITE-BUFFER QUEUES . . . . .	183
D.1	Example of scheduling . . . . .	183
D.2	Scaling limit . . . . .	184
E	QUASI-STEADY STATE APPROXIMATIONS . . . . .	187
E.1	Enzyme-Substrate-Inhibitor System . . . . .	187
F	FUNCTIONAL CENTRAL LIMIT THEOREM . . . . .	191
F.1	Hypergeometric Moments . . . . .	191
F.2	Convergence of the quadratic variation process . . . . .	191
F.3	Interpretation of the $\mathbb{D}$ operator . . . . .	195
G	LUMPABILITY . . . . .	197
G.1	Derivations for local symmetry and fibrations . . . . .	197
H	P2P LIVE STREAMING . . . . .	199
H.1	State-space reduction . . . . .	199

H.2 Mean-field theoretic analysis . . . . .	201
H.3 Game theoretic argument . . . . .	203
NOTATIONS	205
ACRONYMS	207
BIBLIOGRAPHY	209
CURRICULUM VITÆ	225
ERKLÄRUNG LAUT §9 PROMOTIONSORDNUNG	227



## LIST OF FIGURES

---

Figure 1.1	Description of the communication networking scenario . . . . .	2
Figure 1.2	The collaborative uploading scenario . . . . .	4
Figure 2.1	Arrival and service processes of an FJ system. . . . .	12
Figure 2.2	A single-stage queueing system . . . . .	16
Figure 2.3	Description of an MABM . . . . .	20
Figure 3.1	Example of a heterogeneous FJ system. . . . .	22
Figure 3.2	Impact of the degree of usage of a server . . . . .	25
Figure 3.3	Impact of scheduling strategy and parallelisation benefit on the mean waiting time . . . . .	28
Figure 3.4	The impact of the scheduling strategy on the waiting time percentiles. . . . .	29
Figure 3.5	Hierarchical model for heterogeneous FJ systems. . . . .	31
Figure 4.1	Graphical representation of a Markov-additive process . . . . .	34
Figure 4.2	Numerical verification of the bounds for work-conserving systems	39
Figure 4.3	Numerical verification of the bounds for blocking systems. . . . .	46
Figure 4.4	A variant of the round-robin provisioning . . . . .	47
Figure 5.1	The canonical two-path case . . . . .	55
Figure 5.2	Near optimality of proportional allocation and the synchronisation cost . . . . .	57
Figure 5.3	Optimal allocation by minimising regret . . . . .	60
Figure 5.4	The canonical two-path scenario for collaborative stream uploading	61
Figure 6.1	Description of the infinite hypothetical container . . . . .	64
Figure 6.2	Schematic description of the Join-Minimum-Cost (JMC) scheduling.	70
Figure 7.1	Michaelis-Menten kinetics with sQSSA. . . . .	87
Figure 7.2	Michaelis-Menten kinetics with tQSSA. . . . .	91
Figure 7.3	rQSSA for Michaelis-Menten kinetics in the first time scale $\gamma = -2$ .	94
Figure 7.4	rQSSA for Michaelis-Menten kinetics in the second time scale $\gamma = -1$ . . . . .	95
Figure 8.1	Dynamics of the stochastic SI model over a finite time interval. . .	100
Figure 8.2	Comparison of percolation profiles of three degree distributions having the same mean. . . . .	110
Figure 8.3	Comparison of the diffusion approximation with simulation results obtained by Gillespie's algorithm. . . . .	111
Figure 8.4	Time evolution of the correlation coefficient and expected sample paths. . . . .	112
Figure 8.5	Comparison of simulated sample paths. . . . .	113
Figure 8.6	Time evolution of the fraction of nodes on the percolated component and the comparison of cost. . . . .	114
Figure 9.1	Graph fibrations . . . . .	128
Figure 9.2	Lifting procedure to assess the quality of local symmetry-driven lumping. . . . .	130

Figure 9.3	Monotonicity of KL divergence rate . . . . .	134
Figure 9.4	Compression level for local symmetry-based lumping . . . . .	135
Figure 10.1	The buffer as a sliding window. . . . .	140
Figure 10.2	Performance comparison based on mean-field analysis of buffer probabilities. . . . .	150
Figure 10.3	Impact of network structure and performance evaluation. . . . .	152
Figure 10.4	Comparison of different strategy profiles under betweenness centrality-based SCHEDMIX. . . . .	153
Figure 11.1	Summary of contribution . . . . .	156
Figure A.1	Deterministic versus stochastic scheduling . . . . .	169
Figure C.1	Collaborative uploading via three heterogeneous paths . . . . .	180
Figure C.2	The decay rate achieved as a function of the number of packets sent. . . . .	182
Figure H.1	Comparison of all possible strategy vectors in $\mathcal{S}$ . . . . .	204

## LIST OF TABLES

---

Table 4.1	Choices for the Markov-additive process state space for different application scenarios. . . . .	35
Table 7.1	Correspondence between Michaelis-Menten enzyme kinetics and the uploading problem . . . . .	78
Table 7.2	Comparison of conditions for the quasi-steady-state approximations in the stochastic and deterministic Michaelis-Menten (MM) kinetics. . . . .	97



## INTRODUCTION

---

The present dissertation studies two distinct classes of models from applied probability literature and attempts to answer questions that are of asymptotic nature. As a guiding principle, the answers to those questions are obtained by means of various probability approximations. The two classes of models in question belong to the mathematical branches of queueing theory and (stochastic) Interacting Particle System (IPS). Queueing theory finds its most successful applications in operations research and computer science. It possesses a rich literature providing tools for not only Markovian but also non-Markovian processes such as renewal processes. The IPSs, on the other hand, are usually modelled as a continuous time Markov jump process on a configuration space specified by a graph over a collection of particles and a local state space, which is usually assumed to be a compact metric space. In this dissertation, we shall restrict ourselves to local state spaces that are finite. With regard to the graph structure, we shall assume suitable random graph models. In order to put emphasis on the local interaction rules that give rise to a global behaviour, and the role that individual agents (often autonomous) play in the dynamics of the IPS over the configuration space, we shall adopt the name Markovian Agent-based Models (MABMs) for this restricted class of stochastic IPS. The nomenclature is also a deliberate attempt to intertwine the traditional computational Agent-based Models (ABMs) and tools from Markov process literature<sup>1</sup>.

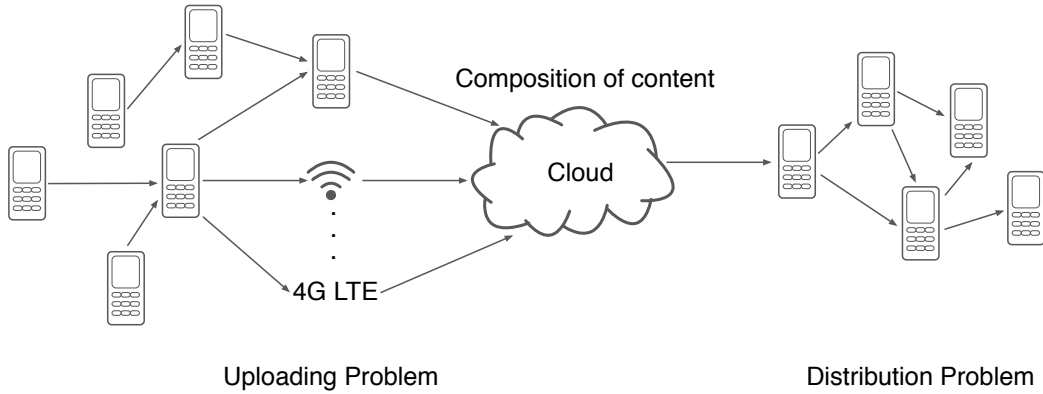
The unlikely marriage of these two branches of probability theory in this dissertation can be largely ascribed to a particular communication networking scenario, the mathematical modelling of which has been a major motivation behind the author's doctoral research work<sup>2</sup>. In the next section, we shall discuss the scenario from a content-centric perspective and use it as a running example throughout the dissertation.

### 1.1 MOTIVATION: A COMMUNICATION NETWORKING SCENARIO

Consider the networking scenario in the context of Internet of Things (IoT) in Figure 1.1. Internet of Things refers to a world of heterogeneous devices, such as sensors and actuators that are connected via various communication technologies while carrying out everyday tasks. On the left hand side of Figure 1.1, we have heterogeneous devices that collaboratively upload certain contents to the cloud. The uploading devices could be smart phones, desktop computers, surveillance cameras, or audio/visual, and ambient sensors. They use neighbouring devices as well as various wired and wireless communication technologies, such as WiFi, cellular, Ethernet and power-line communication, to transmit data. Uploading a picture to the cloud, live streaming using Periscope (Twitter, Inc 2018) or Facebook Live (*Facebook Live* 2018) are some examples. As crowdsourc-

<sup>1</sup> Note that an ABM endowed with an additional Markovian assumption is indeed a special case of the stochastic IPS. Such a Markovian description of an ABM has been adopted by many in the recent times. For instance, see Banisch (2016).

<sup>2</sup> The scenario in question is envisaged in the subproject C3 of the Collaborative Research Centre (CRC) of the German Research Foundation (DFG) Multi-Mechanism Adaptation for the Future Internet (MAKI).



**Figure 1.1:** The description of the communication networking scenario from a content-centric perspective. Heterogeneous devices collaboratively upload content to the cloud. The uploading devices could be smart phones, desktop computers, surveillance cameras, or audio/visual, and ambient sensors. They use neighbouring devices as well as various wired and wireless communication technologies, such as WiFi, cellular, Ethernet and power-line communication, to transmit data. The *uploading problem* is an umbrella term for research problems arising from such a collaborative uploading scenario and encompasses questions such as how long it will take for a data chunk to be transported, how many paths we should choose from among a set of available paths, how to split a data chunk optimally, or if we should allocate or replicate over the different paths available. The cloud in the middle is an abstract representation of a central entity that aggregates, and processes the incoming streams of data to compose new content. The second leg of the scenario concerns distribution of content from the cloud to the end-users. We call it the *distribution problem*. From a mathematical perspective, the distribution problem raises a number of interesting research questions, such as how much time it will take to deliver a certain content to 80 percent of the users, how the graph structure generated by the users impacts the efficiency of the distribution, how we can approximate the system when the number of end-users increases to infinity, or what the scaling limits of such systems are.

ing is gaining traction rapidly (Howe 2006), live events such as music concerts can be covered by composing multiple information streams originating from various mobile devices (Richerzhagen et al. 2016). We use the term crowdsourcing in a broad sense in this dissertation. Crowdsourcing in the context of IoT refers to interconnected devices that ubiquitously exchange and aggregate information to achieve complex goals. The *uploading problem* is an umbrella term for research problems arising from such a collaborative uploading scenario and encompasses questions such as how long it will take for a data chunk to be transported, how many paths we should choose from among a set of available paths, how to split a data chunk optimally, or if we should allocate or replicate over the different paths available. The Chapters 3 to 5 document the scientific contributions pertaining to the uploading problem in adequate generality. Furthermore, the Chapters 6 and 7 are dedicated to the study of two special cases. The cloud in the

middle is an abstract representation of a central entity that aggregates, and processes the incoming streams of data to compose new content<sup>3</sup>.

The second leg of the scenario concerns distribution of content from the cloud to the end-users. We call it the *distribution problem*. Typically, a multicast functionality is realised at the application layer in the form of Content Distribution Networks (CDNs). Often the distribution is made more profitable by introducing Peer-to-Peer (P2P) mechanisms, where the end-users do not rely on the central entity (the cloud in this case) alone and themselves distribute the content among each other (e.g., *BitTorrent* (2018)). From a mathematical perspective, the distribution problem raises a number of interesting research questions, such as how much time it will take to deliver a certain content to 80 percent of the users, how the graph structure generated by the end-users impacts the efficiency of the distribution, how we approximate the system when the number of end-users increases to infinity, or what the scaling limits of such systems are. We explore answers to all such questions pertaining to the distribution leg of the scenario in Chapters 8 to 10.

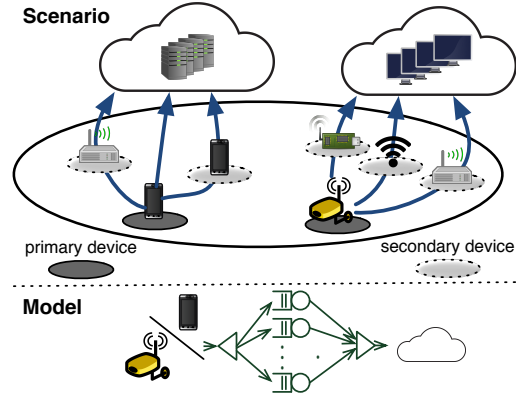
The above scenario will be used as a running example throughout the dissertation. The two key ingredients of the scenario are the uploading and the distribution problems. They are somewhat decoupled, allowing us to pursue them separately. In fact, either of them, to the exclusion of the other, provides us with ample generality to cover adequately many application areas. In order to mathematically model the above scenario, we shall consider various stochastic processes that arise in the context of these two problems, and study their behaviour. The resultant theoretical insights are then exploited to devise *mechanisms* that are optimal in some sense, and also to facilitate *transitions* between them (Frömmgen et al. 2015). As we describe our models in detail in later chapters, we shall actively explore how the tools developed for this communication networking scenario can also be applied to problems in epidemiology, systems and synthetic biology, statistical physics and other branches of science by interpreting the scenario differently.

As a consequence of the inherently decoupled and dissimilar nature of the uploading and the distribution problems, they demand different modelling tools. While the uploading problem is tackled from a queueing theoretic perspective, the distribution problem makes use of the MABMs. We shall make this point more elaborate in the following.

## 1.2 THE UPLOADING PROBLEM FROM A QUEUEING PERSPECTIVE

As described in Section 1.1, the uploading problem concerns heterogeneous devices that collaboratively upload content to the cloud. See Figures 1.1 and 1.2 for different examples of collaborative uploading. The main uploading device is called the primary device and the relaying devices, if any, are called secondary devices. Irrespective of whether there are any secondary devices, the different channels via which the uploading takes place are called *paths*. An important feature of the primary device is *parallelisation*, the ability to simultaneously utilise multiple paths. For instance, in order to upload a photo to the cloud, the primary device can split the file into smaller chunks and upload the chunks using parallel paths. Finally, the photo is reassembled when all of the smaller

<sup>3</sup> The problems related to composition are not considered in this dissertation.



**Figure 1.2:** Collaborative uploading (KhudaBukhsh, Alt, et al. 2018). A device uses neighbouring devices and different paths to upload a data stream.

chunks arrive in the cloud and at that point, we say the uploading is complete. The following are a few other concrete application areas where parallelisation is exploited

1. Multi-path Transmission Control Protocol (Multi-path TCP) (Ford, Raiciu, et al. 2013; Ford and Scharf 2013) splits the data on multiple *subflows* and joins them at the receiver side to ensure in-order data transfer for one logical Transmission Control Protocol (TCP) connection.
2. Recent infrastructural advancement of cloud computing and large-scale data processing has brought about massive deployment of parallel-server systems. Frameworks, such as MapReduce (Dean and Ghemawat 2008; Polato et al. 2014), its implementation Hadoop (Hashem et al. 2016) and Spark (Zaharia et al. 2010) are abundant. Such systems also seek to reap the benefits of parallelisation.
3. Equal-cost Multi-path routing (ECMP)-based (Hopps 2000) load balancing in data-centre networks also distributes packets over multiple paths based on Layer 3 routing information.
4. In Message Passing Interface (MPI)-based parallel computing with master-slave architecture, the master node can be thought of as the primary device that completes a computational task by exploiting several slave nodes simultaneously.
5. In many production houses, different components of a product such as a car (the content in our parlance) are manufactured by independent units simultaneously. The final product is assembled once all of its components are manufactured.

As we see from the examples above from the perspective of the primary device, it is crucial to understand how it can best utilise the parallel paths.

Mathematical modelling of the uploading problem intrinsically involves application-specific challenges. Nevertheless, we look for a useful abstraction that allows us to quantify performance metrics in a meaningful way. With regard to performance metrics for the uploading problem, it is natural to consider quantities such as the time required

to transfer a piece of data from a source device to a processing unit in an edge-cloud, the time to complete a parallel computation task, the amount of time one has to wait before a path becomes available, the total number of packets transported in a given amount of time. Such quantities are the objects of study in queueing theory. Therefore, we adopt queueing theory as a connection-layer abstraction of the uploading problem to mathematically model the performance metrics mentioned above. The analogy becomes clear if we consider the paths to be virtual servers providing service to the data chunks or packets, the “customers” in queueing theoretic language, originating from the primary device. In order to accommodate the “parallel” nature of the uploading problem, we shall put special emphasis on multi-server queueing systems that allow parallelisation.

Besides parallelisation, many of the examples considered above share one more commonality - a synchronisation cost at the output, because the final output is often composed of outputs from all the servers. For instance, in case of MPI-based parallel computation with master-slave architecture, the final computations can be performed only after the slave nodes complete their individual tasks. In case of car manufacturing, the final model can be assembled only after all components have been manufactured by disjoint production units. Therefore, we need to accommodate this output synchronisation constraint as an additional feature in our queueing set-up. Fork-Join (FJ) queueing models naturally capture the dynamics of system parallelisation under synchronisation constraints (Joshi, Soljanin, and Wornell 2017; Rizk, Poloczek, and Ciucu 2015; Thomasian 2014). Since a large number of application areas possess this additional feature, a considerable proportion of queueing theoretic contributions in this dissertation is based on FJ systems, which we describe next.

### 1.2.1 Probability bounds for Fork-Join queueing systems

FJ queueing systems are crucial in the performance evaluation of parallel and distributed systems (Boxma, Koole, and Z. Liu 1994). In an FJ system, arriving jobs are first split into tasks, each of which is then mapped exactly to one work-conserving server that executes the *map* operation. An optional *combine* operation compresses the intermediate result to reduce the amount of data that is transferred through the network. Compression efficiency depends on the application and, in particular, on the input data size. A job finally leaves the system when all of its tasks are executed.

Three key stochastic processes of interest for an FJ system are waiting times, response times, and queue lengths. We define the waiting time for a job to be the amount of time between the arrival of the job and the time when its last task starts getting serviced. That is, a job is said to be waiting until its last task starts being serviced. The response time of a job is the amount of time between the arrival of the job and the time when *all* of its tasks are serviced. Queue length at a server is the number of tasks waiting to be serviced at a particular instant of time. The stochastic behaviour of these processes depends on a number of factors, such as the nature of inter-arrival times of the jobs, *i.e.*, the arrival process; the service times of the servers; the policy of task assignment. It may also depend on extraneous factors that modulate the inter-arrival and the service times.

Although FJ systems are ubiquitous, and form an important class of queueing theoretic models, an exact analysis of FJ systems with more than two servers in a general set-up remains elusive (Baccelli, Makowski, and Shwartz 1989; Boxma, Koole, and Z. Liu

1994). It is particularly hard to find closed-form expressions for the steady-state distributions of the three key stochastic processes described above. One approach to circumvent this problem, which we shall take in this dissertation, is to bound the tail probabilities of the steady-state waiting times (for instance, via a Large Deviations Principle (LDP)). This is precisely our strategy for probability approximation in the context of FJ systems.

From the perspective of performance, it is desirable to have as small waiting times as possible. In order to achieve this objective, a number of optimisation questions arise:

1. How do we model the (application-specific) parallelisation benefit? Consider a Monte-Carlo simulation and a video transcoding application. In the first case, the gain from parallelisation is significant and apparent, while in the second case, it may vary significantly depending on different factors such as the dependency between video macroblocks (Chong et al. 2007; Mesa et al. 2009).
2. Given a model for the parallelisation benefit, how many servers do we choose? Can we achieve satisfactory performance even if we select a subset of the available servers? The decision as to how many servers to choose will be called *scheduling*.
3. How do we optimally divide incoming jobs into tasks? Redundancy techniques have become increasingly popular over the last few years as a tool to decrease latency. While it has been shown to be effective in many cases, how do we in general objectively decide whether to use redundancy? We shall use the term *provisioning* to refer to a rule of job division (into tasks) in this dissertation.

We shall seek answers to these questions in Chapters 3 and 4, where we shall study FJ queueing models in adequate generality allowing for both Markovian and non-Markovian cases. Finally, we shall utilise the tools developed in Chapter 3 and Chapter 4 to devise uploading strategies in Chapter 5.

### 1.2.2 Random time changes of queueing systems

Besides FJ queueing systems, which we adopt for the purpose of modelling parallelisation under synchronisation constraints, we consider a second class of queueing models where no inherent output synchronisation is imposed. In order to model this special class of queueing models, we make use of the random time change representation of the queueing system. In this approach, we characterise the processes of interest such as the queue length as solutions to certain stochastic equations that determine Markov processes (Ethier and Kurtz 1986, Chapter 6). The name “random time changes” can be attributed to the fact that the stochastic equations in question involve a random time change of a second Markov process. Besides being mathematically convenient, this approach is useful for us because it allows for a number of novel approximations and asymptotic results.

The first application of the random time change representation concerns queueing systems with finite buffers, *i.e.*, where the queue length can not grow arbitrarily large. In Chapter 6, we shall document preliminary ideas on optimal scheduling in finite-buffer queueing systems under exogenous modulation. We shall also present a scaling limit as the number of servers increases to infinity in large heterogeneous clusters of finite-buffer servers using the semigroup operator approach to Markov process convergence.

Inspired by chemical physics literature and as the second application of the random time change representation, we carry out Quasi-Steady State Approximations (QSSAs) for the uploading problem when the number of packets to transmit is too large compared to the number of paths available. There is a well established connection between enzyme kinetic Chemical Reaction Networks (CRNs) and queueing systems. We shall make these connections precise in Section 2.3 and Chapter 7. By virtue of the multi-scaling techniques from probability literature (Ball et al. 2006; Kang and Kurtz 2013), we shall derive different variants of QSSAs with a special focus on Michaelis-Menten enzyme-catalysed CRNs in Chapter 7.

### 1.3 THE DISTRIBUTION PROBLEM AS A MARKOVIAN AGENT-BASED MODEL

The distribution problem is modelled as an MABM. Our objective is to distribute a content (or a stream of contents) to a number of end-users, who are interconnected and thereby, form a graph. The distribution is usually facilitated by a P2P-like mechanism in which the end-users themselves *contact* each other and either *pull* or *push* parts of the content (called chunks). In agreement with the ABM parlance, we shall call the end-users *agents*. We assume that each agent maintains a local Poisson clock so that at each ticking of the clock, the agent contacts one of its neighbours and seeks to push/pull a piece of chunk. Additionally, we endow each agent with a local state, which encodes the presence or absence of the chunks of the content. The local state space is assumed finite. Given the above description, the MABM can be seen as a contact process on (random) graphs, and therefore, as a Continuous Time Markov Chain (CTMC) on the configuration space, which is the joint state space of all agents combined, *i.e.*, the Cartesian product of all local state spaces. Precise interaction rules are needed to specify the transition intensities of this Markov chain. One common approach to finding the probability distribution of the Markov chain at a given time point is to solve a set of Ordinary Differential Equations (ODEs), known as the Kolmogorov forward equations in probability literature and Chemical Master Equations (CMEs) in physical sciences.

In order to devise distribution strategies that are optimal in some sense, we need to solve the CMEs. However, there is one major roadblock that needs to be overcome before any strategy optimisation step can be carried out: the size of the configuration space grows exponentially fast with the number of agents. As a consequence, solving the CMEs becomes prohibitively expensive from a computational perspective, and therefore, virtually infeasible. In order to surpass this computational roadblock, we need approximations of the MABM when the number of agents grows large. Therefore, three different approximations are proposed in this dissertation: 1) a diffusion approximation in the form of a Functional Central Limit Theorem (FCLT), 2) an approximately lumpable aggregation of the state space based on a local notion of symmetry (automorphism) of the graph, and finally, 3) a heterogeneous mean-field theoretic approximation.

In Chapter 8, we consider a simple contact process on random graphs, namely an Information-Dissemination (ID) process, which is the same as a stochastic compartmental Susceptible-Infected (SI) process in the epidemiology literature and non-equilibrium percolation in the eyes of a statistical physicist. In the context of the networking scenario presented in Section 1.1, this process captures binary information as to whether the content has reached an agent or not. We prove that, when appropriately scaled, certain



summary statistics (number of agents that already received the content and number of different types of edges in the graph) converge to a Gaussian vector semimartingale as the number of agents increases to infinity, providing us with a scaling limit in the form of a diffusion approximation of the MABM.

In Chapter 9, we consider MABMs in full generality. This time our approximation strategy relies on a local symmetries of the graph. Local symmetries are a generalisation of graph automorphisms. We propose a local symmetry-driven state aggregation strategy that yields approximate lumpability, *i.e.*, the lumped process on the smaller state space is approximately Markovian. Therefore, we can profitably study the reduced system without encountering the computational difficulties posed by the exponentially large configuration space of the original MABM.

In our third approximation, we consider a swarming-based Peer-to-Peer (P2P) live streaming scenario as an application in Chapter 10. We apply heterogeneous mean-field theoretic approximation tools to derive useful recurrence relations among buffer probabilities of the system. The recurrence relations are further utilised to devise a mixed chunk-selection strategy, called SCHEDMIX.

#### 1.4 ORGANISATION OF THE THESIS

The thesis is structured as follows

1. Mathematical preliminaries are provided in Chapter 2. In particular, the basics of FJ queueing systems, the connections between CRNs and queueing systems, lumpability for Markov chains, and MABMs are discussed in this chapter.
2. Chapter 3 presents a stochastic scheduling approach for FJ queueing systems. We provide computable stochastic bounds for the waiting and response time distributions for heterogeneous FJ systems under general parallelisation benefit. The trade-off between the scaling benefit due to parallelisation and the FJ inherent synchronisation penalty is highlighted.
3. Chapter 4 presents an abstract notion of provisioning for FJ queueing systems under changing environments. The changes in the extraneous environment are captured through a Markov additive process. We establish an LDP for the steady-state waiting times, from which computable probability bounds are obtained.
4. Chapter 5 explains how the tools and results obtained in Chapters 3 and 4 can be utilised to devise collaborative uploading strategies. We analyse replication and allocation strategies that control the mapping of data to paths and provide closed-form expressions that pinpoint the optimal strategy given a description of the paths' service distributions.
5. Chapter 6 presents a simple formulation of a queueing system with finite buffers. We present preliminary ideas on how the random time change representation of Markov processes can be used to devise efficient probabilistic scheduling in finite-buffer queueing systems. We also consider heterogeneous, parallel clusters of servers and derive a scaling limit as the number of servers increases to infinity for a class of Join-Minimum-Cost (JMC) scheduling algorithms using the semigroup operator approach to convergence of Markov processes.



6. Chapter 7 presents an application of multi-scaling technique from probability theory literature to a special kind of queueing system that resembles MM enzyme kinetics and derive several QSSAs. In particular, we show how the different assumptions about chemical species abundance and reaction rates lead to the standard QSSA (sQSSA), the total QSSA (tQSSA), and the reversible QSSA (rQSSA). We also illustrate how our approach extends to more complex stochastic networks such as the Enzyme-Substrate-Inhibitor (ESI) system.
7. Chapter 8 presents an FCLT for a stochastic compartmental SI epidemic process on configuration model random graphs with a given degree distribution over a finite time interval. We split the population of graph nodes into two compartments, namely,  $S$  and  $I$ , denoting susceptible and infected nodes, respectively. In addition to the sizes of these two compartments, we study counts of  $SI$ -edges (those connecting a susceptible and an infected node), and  $SS$ -edges (those connecting two susceptible nodes). We show that these counts, when appropriately scaled, converge weakly to a continuous Gaussian vector semimartingale process in the space of vector-valued càdlàg functions endowed with the Skorohod topology.
8. Chapter 9 presents how local symmetries of a graph can be utilised to yield approximate lumpability of an MABM. In a recent paper Simon, Taylor, and Kiss (2011), the authors used the automorphisms of the underlying graph to generate a lumpable partition of the joint state space ensuring Markovianity of the lumped process for binary dynamics. However, many large random graphs tend to become asymmetric rendering the automorphism-based lumping approach ineffective as a tool of model reduction. In order to mitigate this problem, we propose a lumping method based on a notion of local symmetry that compares only local neighbourhoods. The connections to fibrations of graphs are discussed in detail.
9. Chapter 10 explains how mean-field theoretical tools are used to devise a mixed strategy for chunk selection in P2P live streaming applications. We analyse two basic scheduling mechanisms, Latest Deadline First (LDF) and Earliest Deadline First (EDF), and combine them into a mixed strategy, called SCHEDMIX, to leverage inherent differences in client resources. We show that SCHEDMIX outperforms LDF and EDF using a mean-field theoretic analysis of buffer probabilities.
10. Chapter 11 concludes the thesis with a summary of scientific contributions and an outlook for future research.
11. Additional mathematical derivations and supplementary material to the chapters mentioned above are provided at the end of the thesis. In order to facilitate a smooth reading, supplementary materials corresponding to different chapters are presented separately in Appendices A to H.

## 1.5 PUBLICATIONS

We conclude this introductory chapter with a list of articles written during the course of the author's doctoral studies. The following publications are included in parts or in an extended version in this thesis.

## JOURNAL ARTICLES

1. W. R. KhudaBukhsh, C. Woroszylo, et al. (2018). “Functional Central Limit Theorem For Susceptible-Infected Process On Configuration Model Graphs”. In: Submitted.
2. W. R. KhudaBukhsh, S. Kar, et al. (2018). “Provisioning and performance evaluation of parallel systems with output synchronization”. In: Submitted.
3. W. R. KhudaBukhsh, A. Auddy, et al. (2018). “Approximate lumpability for Markovian agent-based models using local symmetries”. In: Submitted.
4. H.-W. Kang, W. R. KhudaBukhsh, et al. (2018). “Quasi-steady-state approximations derived from a stochastic enzyme kinetics model”. In: Submitted.
5. W. R. KhudaBukhsh, J. Rueckert, et al. (2017). “SCHEDMIX: Heterogeneous Strategy Assignment in Swarming-based Live Streaming”. In: Submitted.

## PEER-REVIEWED CONFERENCE PROCEEDINGS

1. W. R. KhudaBukhsh, J. Rückert, et al. (May 2016). “Analysing and leveraging client heterogeneity in swarming-based live streaming”. In: *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, pp. 386–394.
2. W. R. KhudaBukhsh, A. Rizk, et al. (2017). “Optimizing Stochastic Scheduling in Fork-Join Queueing Models: Bounds and Applications”. In: *IEEE International Conference on Computer Communications (INFOCOM)*.
3. W. R. KhudaBukhsh, B. Alt, et al. (Apr. 2018). “Collaborative Uploading in Heterogeneous Networks: Optimal and Adaptive Strategies”. In: *IEEE International Conference on Computer Communications (INFOCOM)*.

**ADDITIONAL PUBLICATIONS** Furthermore, the following publications were part of the author’s PhD research, but, however, are not covered in this thesis. The topics of these publications are outside of the scope of the material covered here.

1. A. Šošić et al. (May 2017b). “Inverse Reinforcement Learning in Swarm Systems (Best Paper Award Finalist)”. In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
2. A. Šošić et al. (May 2017a). “Inverse Reinforcement Learning in Swarm Systems”. In: *AAMAS Workshop on Transfer in Reinforcement Learning*.
3. M. Mousavi et al. (Sept. 2017). “Cross-Layer QoE-Based Incentive Mechanism for Video Streaming in Multi-Hop Wireless Networks”. In: *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pp. 1–7

## PRELIMINARIES

---

In this chapter, we provide necessary mathematical preliminaries and also discuss state of the art. We shall first lay down the notational conventions. We shall then discuss FJ queueing systems. After that, we shall explore the connections between CRNs and queueing systems. Lumpability for both Discrete Time Markov Chain (DTMC) as well as CTMC will be discussed next. Finally, we shall discuss the basic set-up of an MABM.

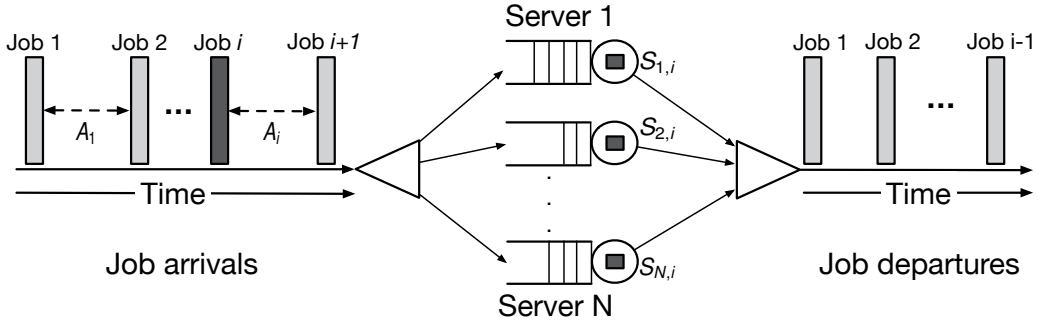
### 2.1 NOTATIONAL CONVENTIONS

The following notational conventions are adhered to throughout the dissertation. We denote the set of natural numbers and the set of real numbers by  $\mathbb{N}$  and  $\mathbb{R}$  respectively. Let  $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ . For  $N \in \mathbb{N}$ , let  $[N] := \{1, 2, \dots, N\}$ . The set of positive real numbers is denoted by  $\mathbb{R}_+$ . For  $F \subseteq \mathbb{R}^N$ , we denote the Borel  $\sigma$ -field of subsets of  $F$  by  $\mathcal{B}(F)$ . For some  $F \in \mathcal{B}(\mathbb{R}^N)$ , the interior, the closure and the boundary of  $F$  are denoted by  $\text{Int } F$ ,  $\text{Cl } F$ , and  $\text{Bnd } F$  respectively. For any extended real-valued function  $f$ , we denote the effective domain of  $f$  by  $\mathcal{D}f$ , i.e.,  $\mathcal{D}f := \{x \in \mathbb{R} \mid f(x) < \infty\}$ . For an event  $F$ , we denote the indicator function of  $F$  by  $\mathbb{1}(F)$ , taking value unity when  $F$  is true and zero otherwise. For a set  $A$ , we denote its cardinality by  $|A|$ , and the class of all subsets of  $A$ , by  $2^A$ . Given  $N, K \in \mathbb{N}$ , the set of all non-negative integer solutions to the Diophantine equation  $x_1 + x_2 + \dots + x_K = N$  by  $\Lambda(N, K)$ , i.e.,  $\Lambda(N, K) := \{x = (x_1, x_2, \dots, x_K) \in \mathbb{N}_0^K \mid x_1 + x_2 + \dots + x_K = N\}$ . The symmetric group on a set  $A$  is denoted by  $\text{Sym}(A)$ .

### 2.2 FORK-JOIN QUEUES

FJ queueing models naturally capture the dynamics of system parallelisation under synchronisation constraints. They have seen a rise of interest as a modelling tool in the wake of massive improvement of the infrastructure for cloud computing and large-scale data processing. The emergence of parallel data processing frameworks such as MapReduce (Dean and Ghemawat 2008; Polato et al. 2014) and its implementation Hadoop (Hashem et al. 2016) has contributed to the modern Information Technology (IT) infrastructure.

We categorise the servers depending on whether they are work-conserving or not. Servers that start servicing the task of the next job, if available, immediately after finishing the current job, are labelled work-conserving. Servers that are not work-conserving, referred to as “blocking” servers hereinafter, wait until *all* servers finish servicing their current tasks before starting the task of the next job. Blocking servers impose an additional synchronisation barrier at the input. We shall show that a blocking system can be treated as a special case of the work-conserving (non-blocking) system. In particular, an FJ system with  $N$  blocking servers can be viewed as a hypothetical queueing system with just one work-conserving server whose service time distribution is the same as the distribution of the maximum order statistic of the individual service times of the  $N$  servers of the original FJ system.



**Figure 2.1:** Arrival and service processes of an FJ system. The random variable  $A_i$  denotes the inter-arrival time between the  $i$ -th and the  $i + 1$ -th jobs. Each incoming job is split into  $N$  tasks and assigned to  $N$  heterogeneous servers. The service time at the  $n$ -th server for the task of the  $i$ -th job is denoted by  $S_{n,i}$ . A job leaves the system when *all* of its tasks are served.

### 2.2.1 Waiting and response times

Consider a single-stage FJ queueing system with  $N$  parallel servers as depicted in Figure 2.1. Jobs arrive at the input station according to some process with inter-arrival time  $A_i$  between the  $i$ -th and  $(i + 1)$ -th job,  $i \in \mathbb{N}$ . A job is split into  $N$  tasks each of which is assigned to exactly one server. The service time for the task of job  $i$  at the  $n$ -th server is denoted by the random variable  $S_{n,i}$ , where  $n \in [N]$  (see Figure 2.1). Finally the job leaves the system when *all* of its tasks are served, imposing a synchronisation constraint at the output. We assume the servers are work-conserving.

We adopt the definition of waiting times and response times from Rizk, Poloczek, and Ciucu (2015). For the first job to arrive, there is no waiting time. For subsequent jobs, we define the waiting time to be the amount of time between the arrival of the job and the time when its *last* task starts getting serviced. That is, a job *waits* until its last task starts getting serviced. The response time is the amount of time between the arrival of a job and the time until *all* tasks of the job are completed. Formally, for an FJ queueing system with  $N$  work-conserving servers, we define the waiting time  $W_j$  for the  $j$ -th job as

$$W_j := \begin{cases} 0 & \text{if } j = 1, \\ \max\{0, \max_{k \in [j-1]} \{\max_{n \in [N]} \{\sum_{i=1}^k S_{n,j-i} - \sum_{i=1}^k A_{j-i}\}\}\} & \text{if } j > 0. \end{cases} \quad (2.2.1)$$

Similarly the response time  $R_j$  of job  $j$  is defined as

$$R_j := \begin{cases} \max_{n \in [N]} S_{n,1} & \text{if } j = 1, \\ \max_{k \in [j-1] \cup \{0\}} \{\max_{n \in [N]} \{\sum_{i=0}^k S_{n,j-i} - \sum_{i=1}^k A_{j-i}\}\} & \text{if } j > 1. \end{cases} \quad (2.2.2)$$

In order to simplify the notations, define the difference process  $Q_k$  (sometimes called the drift process) on the measurable space  $(\mathbb{R}^N, \mathcal{B}(\mathbb{R}^N))$  as follows

$$Q_k := (X_{1,k}, X_{2,k}, \dots, X_{N,k}) \text{ with } X_{n,k} := \sum_{i=1}^k X_{n,i}^A, \quad (2.2.3)$$

where  $X_{n,i}^A = S_{n,i} - A_i$  for all  $i \in \mathbb{N}$  and set  $X_{n,0} := 0$ , for each  $n \in [N]$ . We are interested in the steady-state behaviour of the waiting and the response times. It can be showed that the steady-state waiting time  $W$  and the response time  $R$  have the following distributional representation (see Rizk, Poloczek, and Ciucu (2015)),

$$W \stackrel{\mathcal{D}}{=} \max_{k \in \mathbb{N}_0} \max_{n \in [N]} \{X_{n,k}\}, \quad (2.2.4)$$

$$R \stackrel{\mathcal{D}}{=} \max_{k \in \mathbb{N}_0} \max_{n \in [N]} \{S_{n,0} + X_{n,k}\}, \quad (2.2.5)$$

where  $\stackrel{\mathcal{D}}{=}$  denotes equality in distribution. We need to make some additional technical assumptions to ensure stability of system. We defer a discussion of those technical assumptions to later chapters where we analyse the steady-state waiting and response times in detail. Despite this simple representation, getting closed-form expression of the probability distributions of  $W$  and  $R$  is hard under general settings (Baccelli, Makowski, and Shwartz 1989; Boxma, Koole, and Z. Liu 1994). Useful bounds have been provided in (Baccelli, Makowski, and Shwartz 1989; Balsamo, Donatiello, and Dijk 1998; Rizk, Poloczek, and Ciucu 2015) using probabilistic techniques. Stochastic network calculus has also been used to derive performance upper bounds for FJ systems in (Fidler and Jiang 2016; Kesidis et al. 2015). In this dissertation, we shall also circumvent the difficulty of exact analysis by providing computable bounds on the tail probabilities of the steady-state waiting and response times.

**Remark 2.2.1** (Blocking servers). In many situations the assumption of work-conservingness is not tenable and the servers are “blocking” in nature. This entails forced idleness resulting in higher waiting times. However, as mentioned earlier, we shall show that our framework, although designed for work-conserving systems, is applicable to blocking systems as well by treating an FJ system with  $N$  blocking servers as a virtual queueing system with just one server. In that sense, blocking FJ systems can be analysed within our framework as a special case.

## 2.3 CHEMICAL REACTION NETWORKS AND QUEUEING SYSTEMS

There are interesting analogies between CRNs and queueing systems (see Arazi, Ben-Jacob, and Yechiali (2004) and Gadgil, Lee, and Othmer (2005) and also D. F. Anderson and Kurtz (2011, Chapter 2)). In order to make this analogy clear, let us consider the following example from D. F. Anderson and Kurtz (2011, Example 2.9).

**Example 2.3.1** (M/M/ $\infty$  queues). Consider an M/M/ $\infty$  queueing system, which is a single-stage queueing system with Poissonian arrival, and independent and identically distributed (iid) exponential service times. The term Poissonian arrival refers to the fact that the arrival process is a Poisson process and therefore, the inter-arrival times are iid exponentially distributed with a certain mean. Mathematically, it is equivalent to a birth-death process, where birth refers to the arrival of a customer and death, to the departure after service. The birth rate is constant, while the death rate is proportional to the population size at a given point in time. The M/M/ $\infty$  queueing system can also be

interpreted as a CRN. Intuitively, it is equivalent to a production-degradation reaction, where arrival of a customer refers to the production of a chemical species, and departure, to the degradation of the chemical species. Likewise, the queue length is interpreted as the species copy number (the number of molecules of the species) at a given time. The production-degradation chemical reaction system is schematically described as follows



where  $S$  denotes the chemical species in question. The analogy goes beyond M/M/ $\infty$  queues. Before proceeding with CRN-interpretations of more general queueing systems, we describe the basics of CRNs. A standard textbook is D. F. Anderson and Kurtz (2011).

### 2.3.1 Chemical reaction networks

A CRN is a finite collection of chemical reactions among a finite set of chemical species. Let  $\{S_1, S_2, \dots, S_N\}$  be a set of  $N$  species. Consider the following  $K$  chemical reactions

$$\sum_{n \in [N]} a_{n,j} S_n \rightarrow \sum_{n \in [N]} b_{n,j} S_n, \quad j \in [K]. \quad (2.3.2)$$

where  $a_{n,j}, b_{n,j} \in \mathbb{N}_0$ . That is, in the  $j$ -th reaction,  $a_{n,j}$  molecules of the species  $S_n$  are consumed and  $b_{n,j}$  molecules are produced. When no molecules are consumed or produced in a reaction, *i.e.*, when  $a_{n,j} = 0 \forall j \in [N]$ , or  $b_{n,j} = 0 \forall j \in [N]$ , we simply put the empty set  $\emptyset$ , as we did in case of the production-degradation reaction system in (2.3.1). The quantity  $c_{n,j} := (b_{n,j} - a_{n,j})$  gives the *net* change in the number of molecules of the species  $S_n$  after the  $j$ -reaction. The vector  $c^{(j)} := (b_{1,j} - a_{1,j}, b_{2,j} - a_{2,j}, \dots, b_{N,j} - a_{N,j})$  is called the stoichiometric change vector of the CRN described in (2.3.2).

Let  $X_i(t)$  denote the number of molecules present, the species copy-number, of the  $i$ -th species at time  $t$ , for all  $i \in [N]$ . We are interested in the behaviour of  $X := (X_1, X_2, \dots, X_N)$ . The standard approach is to assume that  $X$  is a CTMC on  $\mathbb{N}_0^N$ . Let  $R_j$  denote the counting process determining the number of times the  $j$ -th reaction has occurred by time  $t$ , for each  $j \in [K]$ . Then,  $X$  satisfies

$$X(t) = X(0) + \sum_{j \in [K]} c^{(j)} R_j(t).$$

We can specify the counting processes precisely by making their intensities explicit. Let  $\lambda_j$  denote the intensity function associated with  $R_j$ , for each  $j \in [K]$ . Then, by the random time change representation for Markov processes (D. F. Anderson and Kurtz 2011; Ethier and Kurtz 1986), the stochastic process  $X$  satisfies the following stochastic equation

$$X(t) = X(0) + \sum_{j \in [K]} c^{(j)} Y_j \left( \int_0^t \lambda_j(X(s)) ds \right),$$

where  $Y_1, Y_2, \dots, Y_K$  are independent, unit Poisson processes. The generator  $\mathcal{A}$  of the Markov process  $X$  is then given by

$$\mathcal{A}f(x) := \sum_{j \in [K]} \lambda_j(x) \left( f(x + c^{(j)}) - f(x) \right),$$

where  $f : \mathbb{N}_0^N \rightarrow \mathbb{R}$  is any given bounded function. Let  $p_t(x) := P(X(t) = x)$  denote the probability distribution of  $X$  at time  $t$ . The time evolution of  $p_t$  is given by the Kolmogorov forward equation, also known as the CME,

$$\frac{d}{dt}p_t(x) = \sum_{j \in [K]} \lambda_j(x - c^{(j)})p_t(x - c^{(j)}) - \sum_{j \in [K]} \lambda_j(x)p_t(x).$$

Now, we specify the intensities  $\lambda_j$ 's.

**MASS-ACTION KINETICS** The most common choice for the intensities  $\lambda_j$ 's is dictated by the *law of mass-action*. When the system is well mixed, each molecule is assumed to be equally likely to react with any other molecule of any species in the system. Therefore, with  $x = (x_1, x_2, \dots, x_N)$ , we set

$$\lambda_j(x) \propto \prod_{n \in [N]} a_{n,j}! \binom{x_n}{a_{n,j}} = \prod_{n \in [N]} (x_n)_{a_{n,j}}.$$

The constants of proportionality, denoted by  $\kappa_j$  for the  $j$ -th reaction, are called the reaction rate constants. Therefore, the mass-action propensities<sup>1</sup> are fully specified by

$$\lambda_j(x) = \kappa_j \prod_{n \in [N]} (x_n)_{a_{n,j}}.$$

Comparing the mass-action propensities  $\lambda_j$ 's, it is now clear that an M/M/ $\infty$  queueing system is indeed equivalent to the production-degradation CRN given in (2.3.1).

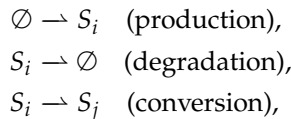
**NON-STANDARD PROPENSITIES** While the mass-action propensities are widely adopted in the literature, there are also other hand-crafted choices that are more appropriate for certain specific application scenarios. Consider the production-degradation reaction system given in (2.3.1). This time we choose the intensities as follows

$$\begin{aligned} \lambda_1(x) &= \kappa_1, \\ \lambda_2(x) &= \kappa_2 \mathbb{1}(x \geq 1), \end{aligned}$$

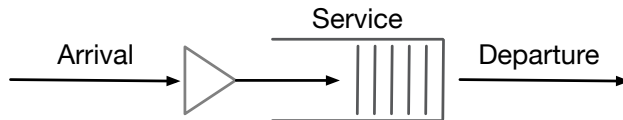
where  $\lambda_1$  is the intensity of production, and  $\lambda_2$  is that of degradation. Note that  $\lambda_2$  does not conform to the mass-action kinetics. However, with this particular choice of the intensities, the production-degradation CRN is equivalent to an M/M/1 queueing system with a First In First Out (FIFO) service routine. See Figure 2.2. We shall explore other examples in the later chapters.

### 2.3.2 First-order reaction networks

First-order reaction networks include conversion-type reactions in addition to production and degradation (Gadgil, Lee, and Othmer 2005). Consider reactions of the form



<sup>1</sup> The terms intensities and propensities are used interchangeably in the CRN literature. They are also often referred to as the reaction hazards.



**Figure 2.2:** A single-stage queueing system. If the arrival process is a Poisson process and if the service times are iid exponential random variables, then the queueing system depicted above is referred to as  $M/M/1$ . Interpreting as a chemical reaction network,  $M/M/1$  queueing system with FIFO service routine is also equivalent to a production-degradation reaction (however, not with mass-action propensity).

for  $i, j \in [N]$ . First-order chemical reaction networks allow us to model a network of  $M/M/\infty$  queues. The correspondence becomes clear with the following analogies: the production of the  $i$ -th species is considered to be an arrival of a customer in the  $i$ -th queueing station; the degradation of the  $i$ -th species is considered to be the departure of a customer from the  $i$ -th queueing station; and finally, a conversion of a molecule of the  $i$ -species to a molecule of the  $j$ -th species is considered to be a transition of a customer from the  $i$ -th queueing station to the  $j$ -th queueing station.

With these analogies in place, we can model any Jackson network with an appropriate first-order CRN. In Arazi, Ben-Jacob, and Yechiali (2004), the authors extend these analogies to the more general class of G-networks, which are a generalisation of the Jackson networks (“G” standing for “generalised”) and model the regulatory circuit responsible for the expression of *lac* operon in *E. coli*.

### 2.3.3 Enzyme kinetics

There is a long-established queueing interpretation of the enzyme kinetic CRNs (Cookson et al. 2011; Hochendoner, Ogle, and Mather 2014; Mather et al. 2011). Enzyme kinetic CRNs are concerned about chemical reactions catalysed by certain enzymes. Let us consider a simple example.

**MICHAELIS-MENTEN ENZYME KINETICS** The MM enzyme-kinetic CRN describes a reversible binding of a free enzyme ( $E$ ) and a substrate ( $S$ ) into an enzyme-substrate complex ( $C$ ), and an irreversible conversion of the complex  $C$  to a product ( $P$ ) and the free enzyme  $E$ . The system is schematically described as follows



In addition to the above reactions, production and degradations of the substrate are also often considered. That is,



Treating the substrates as the “customers” allows us to give a queueing interpretation of the MM enzyme-kinetic CRN. As before, production and degradation of the substrate  $S$  are regarded as the arrival and departure of a customer. The free enzymes  $E$  are regarded as the “servers”. Moreover, we can treat the enzyme-substrate complex  $C$  to



be the “busy” or “occupied” servers. From a biophysics perspective, the “waiting times” for the production of  $P$  are very important. There are several approximations for the analysis of such waiting times.

Intuitively, if the number of free enzymes is large compared to the abundance of the substrate customers (discounting the production and degradation for the moment), there will not be much waiting time for the production of  $P$ . This situation corresponds to what is known as an *underloaded* system in queueing theory. On the other hand, if the abundance of free enzymes  $E$  is small compared to the customers, the corresponding queueing system is considered *overloaded*. In particular, if there are more than one substrates that compete for the same enzymatic service, an underloaded system can engender significant correlation in the production of their respective products. In Cookson et al. (2011), the authors precisely consider such a scenario. Please note that one of the competing substrates can also be an inhibitor. Such CRNs are called Enzyme-Substrate-Inhibitor (ESI) systems and are studied in Chapter 7. Other interesting enzymatic CRNs are analysed from a queueing theoretic perspective in Hochendoner, Ogle, and Mather (2014) and Mather et al. (2011).

## 2.4 LUMPABILITY

We first define lumpability for a DTMC for ease of understanding. We shall later show how the lumpability of a CTMC can be studied using the machinery developed for a DTMC. Standard references on this topic are Buchholz (1994), Kemeny, Snell, et al. (1960), and Rubino and Sericola (1989, 1993).

Let  $\{Y(t)\}_{t \in \mathbb{N}}$  be a time-homogeneous DTMC on a state space  $\mathcal{Y} = [K]$  with transition probability matrix  $T = ((t_{ij}))_{K \times K}$ , where  $t_{ij} := P(Y(2) = j \mid Y(1) = i)$ . Given a partition  $\{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_M\}$  of  $\mathcal{Y}$ , we define a process  $\{Z(t)\}_{t \in \mathbb{N}}$  on  $[M]$  as follows:  $Z(t) = i \in [M] \iff Y(t) \in \mathcal{Y}_i$ , for each  $t \in \mathbb{N}$ . The process  $Z$  is called the *lumped* or the *aggregated* process. The sets  $\mathcal{Y}_i$ ’s are often called lumping classes.

**Definition 2.4.1** (Lumpability of a DTMC). A DTMC  $Y$  on a state space  $\mathcal{Y}$  is said to be lumpable with respect to the partition  $\{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_M\}$  of  $\mathcal{Y}$ , if the lumped process  $Z$  is itself a DTMC for every choice of the initial distribution of  $Y$  (Kemeny, Snell, et al. 1960, Chapter VI, p. 124).

A necessary and sufficient condition for lumpability, known as the Dynkin’s criterion in the literature, is the following: for any two pairs of lumping classes  $\mathcal{Y}_i$  and  $\mathcal{Y}_j$  with  $i \neq j$ , the transition probabilities of moving into  $\mathcal{Y}_j$  from any two states in  $\mathcal{Y}_i$  are the same, i.e.,  $t_{u, \mathcal{Y}_j} = t_{v, \mathcal{Y}_j}$  for all  $u, v \in \mathcal{Y}_i$ , where we have used the shorthand notation  $t_{u, A} = \sum_{j \in A} t_{u, j}$  for  $A \subseteq \mathcal{Y}$ . The common values, i.e.,  $\tilde{t}_{ij} = t_{u, \mathcal{Y}_j}$ , for some  $u \in \mathcal{Y}_i$ , and  $i, j \in [M]$ , form the transition probabilities of the lumped process  $Z$ . Let  $\tilde{T} = ((\tilde{t}_{ij}))_{M \times M}$ . Since the Dynkin’s criterion is both necessary and sufficient, some authors alternatively define lumpability in terms of Dynkin’s criterion. In the literature, the process  $Z$  is sometimes denoted as  $Z = \text{agg}(Y)$ . The following proposition is straightforward.

**Proposition 2.4.1.** *If a DTMC  $Y$  on a state space  $\mathcal{Y}$  is lumpable with respect to a partition  $\{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_M\}$  of  $\mathcal{Y}$ , then the lumped process  $Z$  is a DTMC with transition probability matrix  $\tilde{T}$ . Furthermore, there exists an  $K \times M$  matrix  $V$  such that  $TV = V\tilde{T}$ .*

*Proof.* We define two matrices  $U = ((U_{i,j}))_{M \times K}$  and  $V = ((V_{i,j}))_{K \times M}$  as follows

$$U_{i,j} = \frac{1}{|\mathcal{Y}_i|} \mathbb{1}(j \in \mathcal{Y}_i) \text{ for } i \in [M], j \in [K];$$

$$V_{i,j} = \mathbb{1}(i \in \mathcal{Y}_j) \text{ for } i \in [K], j \in [M].$$

It can be verified that the lumped transition matrix is given by  $\tilde{T} = UTV$ , and by virtue of Kemeny, Snell, et al. (1960, Theorem 6.3.4),  $VUTV = TV$ , which concludes the proof.  $\square$

**Example 2.4.1.** Let  $\mathcal{Y} = \{1, 2, 3\}$  and consider the following transition probability matrix

$$T = \begin{pmatrix} 0.1 & 0.2 & 0.7 \\ 0.2 & 0.1 & 0.7 \\ 0.4 & 0.3 & 0.3 \end{pmatrix}.$$

The DTMC  $Y$  with transition probability matrix  $T$  is lumpable with respect to the partition  $\{\{1, 2\}, \{3\}\}$ , but not with respect to the partition  $\{\{1, 3\}, \{2\}\}$ .

We refer the reader to Buchholz (1994) and Kemeny, Snell, et al. (1960) for further discussion on lumpability. Now, we move to the continuous time case. The lumpability of a CTMC can be equivalently described in terms of lumpability of a linear system of ODEs. Consider the linear system  $\dot{y} = yA$ , where  $A = ((a_{i,j}))$  is an  $K \times K$  matrix (representing the transition rate or the infinitesimal generator matrix of the corresponding continuous time Markov process on state space  $\mathcal{Y} = [K]$ ).

**Definition 2.4.2** (Lumpability of a linear system). The linear system  $\dot{y} = yA$  is said to be lumpable with respect to a partition  $\{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_M\}$  of  $\mathcal{Y}$ , if there exists an  $M \times K$  matrix  $B = ((b_{i,j}))$  satisfying the Dynkin's criterion, i.e., if  $b_{i,j} = \sum_{l \in \mathcal{Y}_j} a_{u,l} = \sum_{l \in \mathcal{Y}_j} a_{v,l}$  for all  $u, v \in \mathcal{Y}_i$ .

The matrix  $B$  is often called a lumping of  $A$ . The following is immediate.

**Proposition 2.4.2.** Consider the linear system  $\dot{y} = yA$  described above. If  $B$  is a lumping of  $A$ , then there exists an  $K \times M$  matrix  $V$  such that  $AV = VB$ .

*Proof.* Similar to the proof of Proposition 2.4.1 and is also provided in Simon, Taylor, and Kiss (2011).  $\square$

As done in the case of a DTMC, we similarly define the lumped process. Notice that the variable  $z = yV$  satisfies the linear system  $\dot{z} = Bz$ , capturing the probability evolution of the lumped system. That is, the matrix  $B$  forms the transition rate matrix of the lumped process.

An alternative approach to study lumpability of a CTMC is via the uniformization of a CTMC. This approach will be particularly useful when we discuss lumpability using local symmetries later. Let us consider a CTMC  $\{Y(t)\}_{t \in \mathcal{T}}$  with transition rate matrix  $A = ((a_{i,j}))$  and some time interval  $\mathcal{T} = [0, T], T > 0$ . The uniformization of  $Y$  entails construction of a DTMC  $\{\tilde{Y}(t)\}_{t \in \mathbb{N}}$  on the same state space with transition probability matrix  $\tilde{A} = ((\tilde{a}_{i,j}))$ , and an independent Poisson process  $\{N(t)\}_{t \in \mathcal{T}}$  with intensity  $m > 0$  constructed in the following way:

1. Choose an  $m$  such that  $m \geq \max\{-a_{i,i} \mid i \in [N]\}$ .
2. Set  $\tilde{a}_{i,j} = a_{i,j}/m$  if  $i \neq j$ , and  $\tilde{a}_{i,i} = 1 + a_{i,i}/m$  otherwise.

A consequence of uniformization is that the original CTMC  $\{Y(t)\}_{t \in \mathcal{T}}$ , and  $\{\tilde{Y}(N(t))\}_{t \in \mathcal{T}}$  are equivalent. In fact, the original CTMC is lumpable with respect to a given partition if and only if the uniformized DTMC is lumpable with respect to the same partition. The uniformized DTMC  $\tilde{Y}$  is often denoted by  $\text{unif}(Y)$ , i.e.,  $\tilde{Y} = \text{unif}(Y)$ . It was proved in Ganguly, Petrov, and Koepl (2014) and Rubino and Sericola (1993) that

$$\text{agg}(\text{unif}(Y)) = \text{unif}(\text{agg}(Y)). \quad (2.4.1)$$

Another useful observation that will be helpful later is regarding permutation of the states. It is intuitive that permutation of elements of the state space does not destroy lumpability of a process. The proof of the following proposition is straightforward, but is provided for the sake of completeness.

**Proposition 2.4.3.** *Let  $Y$  be a CTMC on  $\mathcal{Y}$  with transition rate matrix  $A = ((a_{i,j}))$ . Let  $f \in \text{Sym}(\mathcal{Y})$  be used to permute the states. If  $Y$  (or the linear system  $\dot{y} = yA$ ) is lumpable with respect to a partition  $\{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_M\}$ , then the process  $Z = f(Y)$  is lumpable with respect to the partition  $\{\tilde{\mathcal{Y}}_1, \tilde{\mathcal{Y}}_2, \dots, \tilde{\mathcal{Y}}_M\}$ , where  $\tilde{\mathcal{Y}}_i = \{f(u) \mid u \in \mathcal{Y}_i\}$ .*

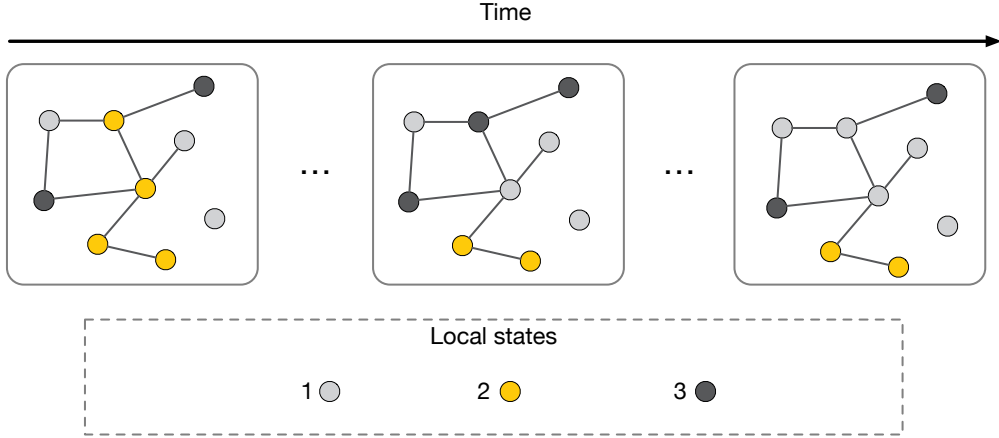
*Proof of Proposition 2.4.3.* It can be verified that  $\{\tilde{\mathcal{Y}}_1, \tilde{\mathcal{Y}}_2, \dots, \tilde{\mathcal{Y}}_M\}$  indeed forms a partition of  $\mathcal{Y}$ . Let us denote the transition rate matrix of  $Z$  by  $\tilde{A} = ((\tilde{a}_{i,j}))$ , where  $\tilde{a}_{i,j} = a_{f^{-1}(i), f^{-1}(j)}$ , and  $f^{-1}$  is the inverse of  $f$  in  $\text{Sym}(\mathcal{Y})$ . The proof will be complete if we show that the linear system  $\dot{z} = z\tilde{A}$  is lumpable. Pick  $\tilde{\mathcal{Y}}_i$ , and  $\tilde{\mathcal{Y}}_j$  for  $i \neq j$ , and let  $u, v \in \tilde{\mathcal{Y}}_i$  be arbitrarily chosen. See that  $u \in \tilde{\mathcal{Y}}_i$  implies  $f^{-1}(u) \in \mathcal{Y}_i$ . Then,

$$\sum_{l \in \tilde{\mathcal{Y}}_j} \tilde{a}_{u,l} = \sum_{l \in \tilde{\mathcal{Y}}_j} a_{f^{-1}(u), f^{-1}(l)} = \sum_{l \in \mathcal{Y}_j} a_{s,l} = \sum_{l \in \mathcal{Y}_j} a_{t,l} = \sum_{l \in \mathcal{Y}_j} a_{f^{-1}(v), f^{-1}(l)} = \sum_{l \in \tilde{\mathcal{Y}}_j} \tilde{a}_{v,l},$$

where  $s = f^{-1}(u), t = f^{-1}(v) \in \mathcal{Y}_i$  and the equality for  $s$  and  $t$  holds by virtue of the lumpability of  $Y$ . This verifies the Dynkin's criterion for  $\dot{z} = z\tilde{A}$ .  $\square$

## 2.5 MARKOVIAN AGENT-BASED MODELS

The MABMs are a marriage of two dissimilar and mature disciplines. Given a graph with  $N$  vertices, we endow each vertex with a local state that varies over time stochastically as the vertex interacts with its neighbours. For instance, in Figure 2.3, each vertex is endowed with a local state that takes values in  $\{1, 2, 3\}$  (depicted in grey, yellow and black). The vertices change colour as they interact with their direct neighbours. The interaction rules are application-specific, and so is the physical interpretation of the vertices of the graph. The MABMs arise naturally in epidemiology, statistical physics, computer science, biology, and engineering disciplines. The simplest example of an MABM is the SI process from epidemiology literature. The objective of the SI model is to study the spread of an infectious disease over a human population. From the perspective of a computer scientist, the SI process is an ID process. The SI process can also be employed to model the spread of a computer virus over a computer network. In the eyes of a statistical physicist, the SI process describes non-equilibrium percolation.



**Figure 2.3:** Illustration of a dynamical process on a graph. Each node is endowed with a local state space  $\{1, 2, 3\}$ , shown in three different colours. They change their colours as a result of local interactions.

Classical modelling approaches to the subject usually ignore the graph structure, and assume some sort of “homogeneous mixing” in the sense that any individual (assuming the vertices represent individuals) can interact with any other individual. Mathematically, this amounts to assuming the underlying graph is a complete graph. This assumption simplifies the analysis of such processes. For instance, mean-field techniques from statistical physics also rely on this assumption. However, this assumption is not justified for most practical applications. In fact, different applications may demand dedicated models for the graph itself to exhibit appropriate structures and properties.

Let  $G = (V, E)$  be a graph (possibly a realisation of a random graph), where  $V = [N] := \{1, 2, \dots, N\}$  is the set of vertices, and  $E \subseteq V \times V$  is the set of edges. Let  $X_i(t)$  denote the local state of vertex  $i \in [N]$  at time  $t \in \mathcal{T} := [0, T]$  for some  $T > 0$ . For simplicity, we assume the vertices have the same finite local state space  $\mathcal{X}$ , *i.e.*,  $X_i \in \mathcal{X}$ , for all  $i \in [N]$ . We are interested in the process  $X := (X_1, X_2, \dots, X_N) \in \mathcal{X}^N$ . We assume the process  $X$  is a CTMC, whose transition rates depend on  $G$ .

Given the CTMC formulation, the time evolution of the probability distribution  $p_t$  of  $X$  at time  $t$  can be described by the Kolmogorov forward equations (W. J. Anderson 1991), a set of ODEs. This makes for a principled approach, but the size of the state space  $\mathcal{X}^N$  grows exponentially as  $N$  grows large. As a result, solving the ODEs is computationally infeasible when  $N$  is large. Therefore, we shall consider various approximations for the MABMs in Chapters 8 to 10.

In this chapter, we study stochastic scheduling for FJ systems. In doing so, we model one of the main advantages of parallel systems, namely, the application specific parallelisation benefit. To this end, we use the notion of service time scaling at each server of the FJ system. However, since a job can only leave the system when all of its tasks are executed, we observe a naturally arising synchronisation penalty in FJ systems. We analytically highlight this trade-off for arbitrary parallelisation benefit regimes. We also show the impact of heterogeneous servers on this trade-off.

Since in large pools of cloud resources, and in many parallelised systems, jobs are not mapped to *all* available resources, and given the performance trade-off mentioned above, it is crucial to select the number of servers to utilise from a given pool of available servers in an informed way. In the context of FJ systems, we define a scheduling strategy to be a probability distribution over the number of available servers. A deterministic strategy is hence a degenerate case. We shall make these ideas precise in the following.

### 3.1 HETEROGENEOUS FORK-JOIN QUEUEING SYSTEMS

#### 3.1.1 System description

Consider a single-stage FJ queueing system with  $N$  parallel servers as described in Section 2.2 (also see Figure 2.1). We shall assume independence of the families of service and inter-arrival times  $\{S_{n,i}\}$  and  $\{A_i\}$  throughout this chapter.

We assume that the families  $\{S_{n,i}\}$  and  $\{A_i\}$  admit finite Moment Generating Function (MGF) and Laplace transform, defined as  $\alpha_n(\theta) := E[\exp(\theta S_{n,1})]$ ,  $\beta(\theta) := E[\exp(-\theta A_1)]$ , respectively, for some  $\theta > 0$  and for all  $n \in [N]$ . We assume the service times are iid. In addition to that, we also assume the job arrival process is a renewal process, *i.e.*, the inter-arrival times are also assumed iid. Finally, we assume the stability condition  $E[S_{n,1}] < E[A_1]$  for each  $n \in [N]$ .

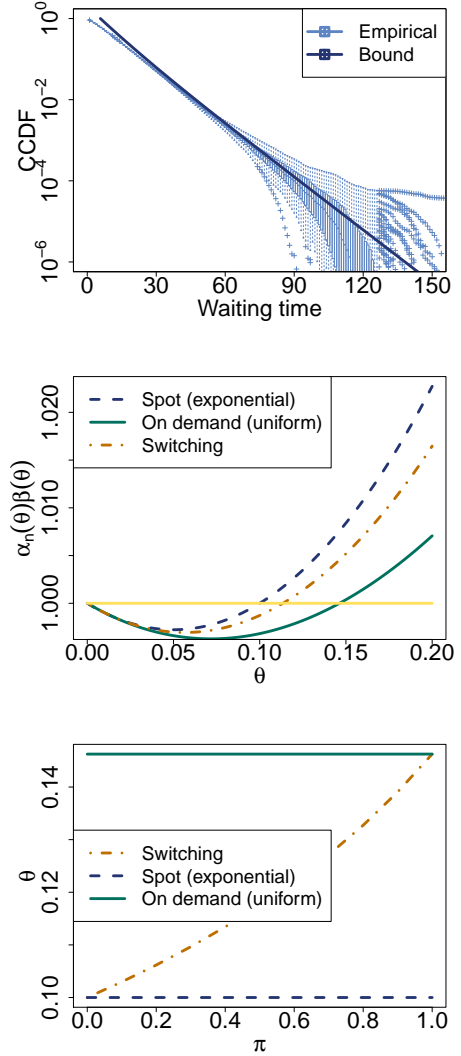
#### 3.1.2 Waiting and response times for heterogeneous FJ Systems

As defined in Section 2.2, the steady-state waiting and response times are given by

$$\begin{aligned} W &\stackrel{\mathcal{D}}{=} \max_{k \in \mathbb{N}_0} \left\{ \max_{n \in [N]} \left\{ \sum_{i=1}^k S_{n,i} - \sum_{i=1}^k A_i \right\} \right\}, \\ R &\stackrel{\mathcal{D}}{=} \max_{k \in \mathbb{N}_0} \left\{ \max_{n \in [N]} \left\{ \sum_{i=0}^k S_{n,i} - \sum_{i=1}^k A_i \right\} \right\}. \end{aligned} \tag{3.1.1}$$

It is infeasible to find the probability distributions of  $W$  and  $R$  in closed-form for arbitrary laws of the inter-arrival and service times (see Section 2.2). Therefore, we provide the following bounds on the tail probabilities of  $W$  and  $R$ .

**Figure 3.1:** Example of a heterogeneous FJ system. **(Top)** Waiting time performance in a MapReduce cloud scenario with  $N = 2$  partially volatile servers. One server is on an average faster representing a revocable checkpointed spot server with an exponential tail of service time. The second server provides on average slower service with uniformly distributed service times representing an on-demand server with stronger guarantees. The bound is calculated using Theorem 3.1.1. The  $y$ -axis denotes the Complementary Cumulative Distribution Function (CCDF). **(Middle)** The FJ system is constrained by the (on an average) faster spot server due to its larger higher-order moments. This is apparent in the MGF condition  $\alpha_n(x)\beta(x) = 1$ . Observe that the constraining decay rate is given by  $\tilde{\theta} := \min_{n \in [N]} \theta_n$ . **(Bottom)** A system that switches between spot and on-demand servers with  $\pi$  being the fraction of time where on-demand servers are used. Observe the improvement in the decay rate  $\theta$  with increasing  $\pi$ . Simulation parameters: spot exponential service rate  $\mu = 1$ , inter-arrival exponential rate  $\lambda = 0.9$  and uniform service time over  $[0.001, 2.009]$ .



**Theorem 3.1.1.** Consider a stable FJ system with  $N$  parallel work-conserving servers fed by renewal job arrivals with inter-arrival times  $A_i$ , for  $i \in \mathbb{N}$ . Assuming iid service times  $S_{n,i}$  and pairwise independence of the servers, the tail probabilities of the steady state waiting and response times are bounded by

$$P(W \geq \sigma) \leq \exp(-\tilde{\theta}\sigma) \sum_{n \in [N]} \exp(-(\theta_n - \tilde{\theta})\sigma),$$

$$P(R \geq \sigma) \leq \exp(-\tilde{\theta}\sigma) \sum_{n \in [N]} \alpha_n(\theta_n) \exp(-(\theta_n - \tilde{\theta})\sigma),$$

where  $\theta_n > 0$  is such that  $\alpha_n(x)\beta(x) = 1$  for  $n \in [N]$  and  $\tilde{\theta} := \min_{n \in [N]} \theta_n$ .

The key steps involved in the proof of the above theorem are: 1) constructing separate martingales for each of the servers; and 2) applying Doob's sub- and supermartingale

inequalities (see Ash (1972) and Durrett (2010a)) to arrive at the bounds. The detailed proof is provided in Appendix A. Note that the stability condition guarantees the existence of  $\theta_n > 0$  such that  $\alpha_n(\theta_n)\beta(\theta_n) = 1$  for all  $n \in [N]$  (see Boxma, Koole, and Z. Liu (1994) and Poloczek and Ciucu (2014)). Hence,  $\hat{\theta} > 0$  is well defined.

**Example 3.1.1** (Hedging using revocable cloud resources). We consider a mixed cloud service consisting of both highly guaranteed and revocable resources. This service could be supplied by infrastructure providers such as *Amazon Elastic Compute Cloud EC2* (2016), or by a virtual provider on top using, e.g., on-demand or revocable spot market machines (Subramanya et al. 2015).

Consider an application of parallel computation under synchronisation such as MapReduce (*Amazon Elastic Compute Cloud EC2* 2016) or Spark (*Apache Spark* 2016) requiring  $N$  machines. In this example, we consider the case of exchanging on-demand machines with spot machines to save cost. In general, for a fixed budget the user obtains *faster* spot machines in comparison to on-demand machines. The price difference arises naturally since spot machines are at a risk of revocation (Subramanya et al. 2015). In order to mathematically model the abstract characteristics of these two classes of machines (*on-demand* and *spot*), we use different job service time distributions. Through revocation and application checkpointing procedures (Subramanya et al. 2015) that are associated with spot machines, we generally make the tail of the corresponding job service time distributions to decay slower than in the case of on-demand machines. For illustration we assume that the tail of the job service times decays exponentially in case of spot machines while in the case of on-demand machines we model the service times by a uniform distribution. Note that the specific choice of the distributions is immaterial for the argument as long as the tail of the service times decays slower for spot machines.

Figure 3.1 (left) shows the waiting time distribution in the case of exchanging an on-demand machine by an - on an average faster - spot machine. At first sight this seems to be a good idea, however, looking at Figure 3.1 (middle) we clearly see that the system is constrained by the spot machine, which has a lower average service time, but also has a thicker tail. The figure on the right shows the utility of trading an on-demand machine with a spot one. While a greater usage of the on-demand machine incurs greater cost, it also increases the decay rate of the waiting and response times,  $\theta$  which in turn leads to monetary saving due to faster job execution times.

### 3.2 SCHEDULING TASKS IN HETEROGENEOUS FJ SYSTEMS

In this section, we study basic scheduling mechanisms that decide on the number of servers to be used from a pool of available servers<sup>1</sup>. Since in large pools of cloud resources (in general, for any parallelised system) an arriving job is not scheduled on *all* available resources, we assign a probability to each of the servers to decide whether a server is to be selected to execute the task of an arriving job. Specifically, when a job arrives, the  $n$ -th server is selected with a probability  $\pi_n$ . This server selection probability  $\pi_n$  can be used to model different aspects of parallelised systems, such as the server failure rate in cloud computing facilities, a quality of service differentiation parameter

<sup>1</sup> Note that our notion of scheduling differs from traditional scheduling algorithms such as the Shortest Remaining Processing Time (SRPT)-first.



for different applications, and a tuning parameter to control the degree of replication. Hence, different  $\pi_n$  may exist for different classes of users. Mathematically, the revised task service times  $\tilde{S}_{n,i}$  are defined as  $S_{n,i}$  with probability  $\pi_n$  and 0 with probability  $1 - \pi_n$ . The MGF of  $\tilde{S}_{n,i}$  is given by  $\alpha_n^*(\theta) = (1 - \pi_n) + \pi_n \alpha_n(\theta)$ . The stability condition  $\max_{n \in [N]} \mathbb{E}[S_{n,1}] < \mathbb{E}[A_1]$  ensures the existence of the decay rate  $\theta_n > 0$  from Theorem 3.1.1 for each  $n \in [N]$  such that  $\alpha_n^*(\theta_n)\beta(\theta_n) = 1$ . Define  $\tilde{\theta} := \min_{n \in [N]} \theta_n > 0$ . We retain the same mathematical set-up as before except for  $S$  being replaced by  $\tilde{S}$ .

**Theorem 3.2.1.** *Consider a stable FJ system with  $N$  parallel work-conserving servers fed by renewal job arrivals with inter-arrival times  $A_i$ , for  $i \in \mathbb{N}$ . The probability that the  $n$ -th server is selected at the arrival of a job is  $\pi_n$ . Assuming iid service times  $S_{n,i}$  and pairwise independence of the servers, the tail probabilities of the steady state waiting and response times are bounded by*

$$\begin{aligned} P(W \geq \sigma) &\leq \exp(-\tilde{\theta}\sigma) \sum_{n \in [N]} \exp(-(\theta_n - \tilde{\theta})\sigma), \\ P(R \geq \sigma) &\leq \exp(-\tilde{\theta}\sigma) \sum_{n \in [N]} \alpha_n(\theta_n) \exp(-(\theta_n - \tilde{\theta})\sigma), \end{aligned}$$

where  $\theta_n > 0$  is such that  $\alpha_n^*(x)\beta(x) = 1$ , for  $n \in [N]$  and  $\tilde{\theta} := \min_{n \in [N]} \theta_n$ .

The proof is similar to that of Theorem 3.1.1. However, for the sake completeness, it is provided in Appendix A.

**Example 3.2.1** (Mixed server pool with different availability). Consider a pool of heterogeneous servers that are available according to some probability  $\pi_i$ . For simplicity, we consider only three heterogeneous servers for parallel processing. Note that this scenario can be easily generalised to  $N$  servers using Theorem 3.2.1. For the sake of simplicity, we assume that the task service times are exponentially distributed with server specific rates  $\mu_i$  and that jobs arrive according to a renewal process with exponentially distributed inter-arrival times with parameter  $\lambda$ . Note that the probability  $\pi_i$  also signifies the fraction of time server  $i$  is used, hence, it is directly related to the computation cost in case of time priced resources.

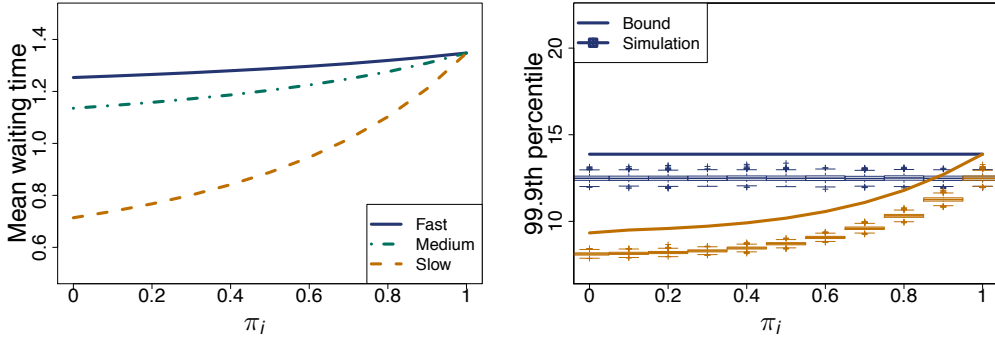
Figure 3.2 shows the change in the mean and the percentile of the waiting time due to the addition of a server with a selection probability  $\pi_i$  to a system of two permanently used servers each with  $\pi_j = 1$ . For example, the lowest curve in Figure 3.2 (left) shows the increase in the average waiting time if the slowest server is added with increasing probability  $\pi_i$ .

**OPTIMAL STRATEGY** It can be shown that the bound in Theorem 3.2.1 is an increasing function of the number of servers  $N$  and that the decay rate  $\tilde{\theta}$  can be maximised, i.e., the bound can be minimised by choosing only the the strongest server.

### 3.3 SCHEDULING UNDER APPLICATION SPECIFIC SCALING

In this section, we analyse scheduling in FJ systems under application specific workloads. We build on the fact that different applications receive different gains from parallelisation that is inherent to the application itself. Consider a Monte-Carlo simulation and a video transcoding application. In the first case, the gain from parallelisation is significant





**Figure 3.2:** Impact of the degree of usage of a server on the mean (left) and the 99.9-th percentile (right) of the steady-state waiting times. We consider a pool of three heterogeneous servers (fast, medium, slow), where tasks are always scheduled on two servers and the third server is included with probability  $\pi_i$ . Parameters: service exponential rates  $(\mu_1, \mu_2, \mu_3) = (1.5, 1.25, 1)$  and inter-arrival exponential rate  $\lambda = 0.5$ .

and apparent, while in the second case, the gain from parallelisation may vary depending on different factors such as the dependency between video macroblocks (Chong et al. 2007; Mesa et al. 2009). We capture these varying gains using the notion of scaled service times. Moreover, in FJ systems (e.g., MapReduce) there is a synchronisation price that increases with the number of servers  $N$  (Baccelli, Makowski, and Shwartz 1989; Rizk, Poloczek, and Ciucu 2015). We argue that given these two opposing forces, the scheduling strategy that chooses the number of servers to utilise in an FJ system can be optimised to minimise the waiting and response times in the system. We begin with the simple case of homogeneous servers before discussing the more general case of heterogeneous servers.

### 3.3.1 Homogeneous Servers - Linear scaling

The first natural scaling that we analyse is what we call *linear scaling*<sup>2</sup>. This is motivated by examples of FJ systems where incoming jobs are equally divided among the servers. Consider an FJ system with  $N$  parallel, identical servers fed by renewal job arrivals with inter-arrival times  $A_i$ . We choose the servers probabilistically and once chosen, stick to them for a long time. This allows us to write down steady-state representations conditional on the chosen set. Let the random variable  $L \sim f_L \in \mathcal{P}([N], 2^{[N]})$  denote the number of servers chosen to split an incoming job into, where  $\mathcal{P}([N], 2^{[N]})$  is the class of all probability distributions on the measurable space  $([N], 2^{[N]})$ . Let the service times at the  $n$ -th server  $S_{n,i}$  be iid for all  $i \in \mathbb{N}$  and  $n \in [N]$ . Suppose the unscaled service time at each server is distributed as  $S$ , i.e.,  $S_{n,i} \mid \{L = 1\} \stackrel{\mathcal{D}}{=} S$  for some  $S$  with MGF  $\alpha(x)$ . We

<sup>2</sup> Linear scaling has been introduced in Rizk, Poloczek, and Ciucu (2016) for a fixed number of homogeneous servers  $N$  without considering scheduling strategies.

model the reduction of the amount of work to be performed by each server when we use  $l$  servers using the following scaling of service times

$$S_{n,i} \mid \{L = l\} \stackrel{\mathcal{D}}{=} \frac{S}{l}. \quad (3.3.1)$$

Now, conditional on the given number of used servers  $\{L = l\}$  for some  $l \in [N]$ , the steady-state waiting times  $W$  and the response times  $R$  can be represented as in (3.1.1) with  $[N]$  replaced by  $[l]$ . We have the following result.

**Theorem 3.3.1.** *Consider a stable FJ system with  $N$  parallel work-conserving servers and renewal job arrivals with inter-arrival times  $A_i$ , for  $i \in \mathbb{N}$ . Let  $L \sim f_L \in \mathcal{P}([N], 2^{[N]})$  denote the number of servers chosen to split an incoming job into. Let the unscaled service times  $S$  and the inter-arrival times  $A$  be exponentially distributed with parameters  $\mu$  and  $\lambda$ , respectively. For service times  $S_{n,i}$  at the  $n$ -th server that are scaled as in (3.3.1) independently for all  $n \in [L]$ ,  $i \in \mathbb{N}_0$ , the tail probabilities of the steady state waiting and response times are bounded as*

$$\begin{aligned} P(W \geq \sigma) &\leq e^{\lambda\sigma} E[Le^{-\mu\sigma L}], \\ P(R \geq \sigma) &\leq \frac{e^{\lambda\sigma}}{\rho} E[L^2 e^{-\mu\sigma L}], \end{aligned}$$

where  $\rho = \frac{\lambda}{\mu}$  is the unscaled utilisation level and the optimal strategy with respect to the bound for the waiting time is

$$L_{opt} \sim f_{opt} = \underset{f_L \in \mathcal{P}([N], 2^{[N]})}{\operatorname{argmin}} E[Le^{-\mu\sigma L}].$$

The proof is provided in Appendix A. For a given choice of the distribution of  $L$ , which we call a *strategy*, the bounds in Theorem 3.3.1 can be computed exactly, for it involves a summation of only finitely many terms. Note that the optimisation is essentially over a probability  $N$ -simplex  $\Delta_N := \{(p_1, p_2, \dots, p_N) \in [0, 1]^N \mid \sum_{k=1}^N p_k = 1\}$ .

**Remark 3.3.1** (Interpretation of the server selection strategy). A strategy can be interpreted in two ways: (i) it actively arises through users' selection of different numbers of servers to utilise, or (ii) it passively arises through a variable number of provided servers that are price volatile, e.g., spot instances at a given budget. In the following, we mainly take the former as an example for strategy derivations.

Note that different strategies lead to varying performance bounds, e.g., consider the case where we select the number of used servers uniformly at random from the pool of  $N$  servers, i.e.,  $P(L = l) = (1/N)\mathbb{1}(l \in [N])$ . Then, for  $a > 0$ ,

$$\begin{aligned} E[Le^{-aL}] &= \frac{e^{-a}}{N(1 - e^{-a})} \left[ \frac{1 - e^{-(N+1)a}}{(1 - e^{-a})} - (N+1)e^{-aN} \right], \\ E[L^2 e^{-aL}] &= \frac{e^{-2a}}{N(1 - e^{-a})} \left[ 2 \frac{(1 - e^{-(N+1)a})}{(1 - e^{-a})^2} - \frac{2(N+1)e^{-Na} - (1 - e^{-(N+1)a})}{(1 - e^{-a})} \right. \\ &\quad \left. - (N+1)(Ne^{-(N-1)a} + e^{-aN}) \right]. \end{aligned}$$

Setting  $a = \mu\sigma$ , closed-form expressions for the bounds in Theorem 3.3.1 are obtained. The uniform distribution allows little control over the number of selected servers. In order to control the average number of utilised servers  $E[L]$  we employ what we call a *Binomial strategy*, i.e., we let  $L$  follow a truncated binomial distribution on  $[N]$  with parameters  $N$  and  $p \in (0, 1]$ ,

$$P(L = l) = \frac{\binom{N}{l} p^l q^{N-l}}{1 - q^N} \mathbb{1}(l \in [N]),$$

where  $q := 1 - p$ . With abuse of notation, we write  $L \sim \text{Binomial}(N, p)$ . Given the total number of available servers  $N \in \mathbb{N}$ , the binomial strategy allows us to vary  $p$  to control the desired average number of utilised servers  $Np/(1 - q^N)$ .

Computing the expectations in Theorem 3.3.1 for  $L \sim \text{Binomial}(N, p)$ , we get the following bounds

$$P(W \geq \sigma) \leq Ne^{-\theta\sigma} \left[ \frac{p}{1 - q^N} (pe^{-\mu\sigma} + q)^{N-1} \right] \quad (3.3.2)$$

$$P(R \geq \sigma) \leq \frac{Ne^{-\theta\sigma}}{\rho} \left[ \frac{p}{1 - q^N} (Npe^{-\mu\sigma} + q)(pe^{-\mu\sigma} + q)^{N-2} \right].$$

The derivation is provided in Appendix A.

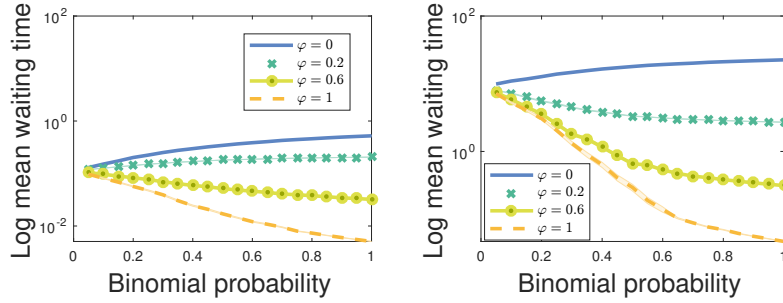
**OPTIMISING THE BINOMIAL STRATEGY** Our next goal is to minimise the tail probabilities of the steady-state waiting times given a binomial strategy for server selection. Precisely, given  $N$  available servers we look for  $p$  that minimises the right hand side of (3.3.2) at some percentile  $\sigma$ , e.g., the 99.9-th percentile. First, we rewrite the right hand side of (3.3.2) as

$$Ne^{-\theta\sigma} \left[ (\epsilon q + 1 - \epsilon)^{N-1} / \sum_{k=0}^{N-1} q^k \right],$$

where we define  $\epsilon := 1 - \exp(-\mu\sigma)$ . Next, we define  $\psi : [0, 1) \rightarrow \mathbb{R}_+$  as  $\psi(q) := (\epsilon q + 1 - \epsilon)^{N-1} / \sum_{k=0}^{N-1} q^k$  and study its behaviour. Taking derivative with respect to  $q$ , we get

$$\frac{d}{dq} \psi(q) = \frac{(\epsilon q + 1 - \epsilon)^{N-2}}{(\sum_{k=0}^{N-1} q^k)^2} \sum_{k=0}^{N-2} (N\epsilon - 1 - k) q^k.$$

Since  $(\epsilon q + 1 - \epsilon)^{N-2} / (\sum_{k=0}^{N-1} q^k)^2 > 0$ , the sign of the derivative is dictated by sign of the polynomial  $Q(q) := \sum_{k=0}^{N-2} (N\epsilon - 1 - k) q^k$ . Note that the coefficients  $\{N\epsilon - 1 - k\}_{k \in \{0\} \cup [N-2]}$  of the polynomial are monotonically decreasing, implying there is only one change of sign of the coefficients so that by *Descartes' rule of signs*, there is at most one real root of  $Q(q) = 0$ . Consequently, the same holds true for  $\frac{d}{dx} \psi(x)$ . Now, observe that  $Q(0) = N\epsilon - 1 > 0$  if  $\epsilon > 1/N$ . On the other hand,  $Q(1) = N(N-1)(\epsilon - 1/2) > 0$  if  $\epsilon > 1/2 \iff \sigma > (1/\mu) \ln(2)$ . This condition on the 99.9-th percentile of the waiting time holds except for corner cases with nearly no queueing. This gives us a sufficient condition for  $\frac{d}{dx} \psi(x) > 0$  implying that  $\psi(q)$  is an increasing function of  $q$  on  $\epsilon > 1/2$ . In other words, the tail bound is a decreasing function of  $p$ . Therefore, the optimal strategy would be to set  $p_{\text{opt}} = 1$  and use all  $N$  available servers to make the most of the scaling benefit. Our analytic arguments are also numerically validated using simulations in Figure 3.4.



**Figure 3.3:** The impact of the scheduling strategy (given by probability  $p$ ) together with the parallelisation benefit (given by increasing  $\varphi$ ) on the mean waiting time in given FJ systems. Simulation parameters:  $N = 10$  servers, (Left) low utilisation:  $\lambda = 0.1$ . (Right) high utilisation:  $\lambda = 0.9$ .

**OPTIMISATION UNDER BUDGET CONSTRAINT** In the interesting scenario of an application with a budget constraint on the average number of servers it uses, the above reduces to a constrained optimisation problem. Precisely, if we have a budget constraint of the form  $E[L] \leq L^*$ , the optimisation problem can be stated as

$$\min N e^{-\theta\sigma} \left[ \frac{p}{1-q^N} (p e^{-\mu\sigma} + q)^{N-1} \right] \quad \text{such that} \quad \frac{Np}{1-q^N} \leq L^*,$$

leading to  $p^* = \sup\{p \in (0, 1] \mid \sum_{k=0}^{N-1} (1-p)^k \geq \frac{N}{L^*}\}$  so that  $f_{opt} = \text{Binomial}(N, p^*)$ . In general, the given bound can always be numerically optimised for any  $\sigma$ .

**GENERALISATION TO POWER SERIES STRATEGIES** In order to obtain bounds in the more general set-up of a power series strategy, we assume

$$P(L = l) := \frac{a_l \kappa^l}{\zeta(\kappa)} \mathbb{1}(l \in \mathbb{N}), \quad (3.3.3)$$

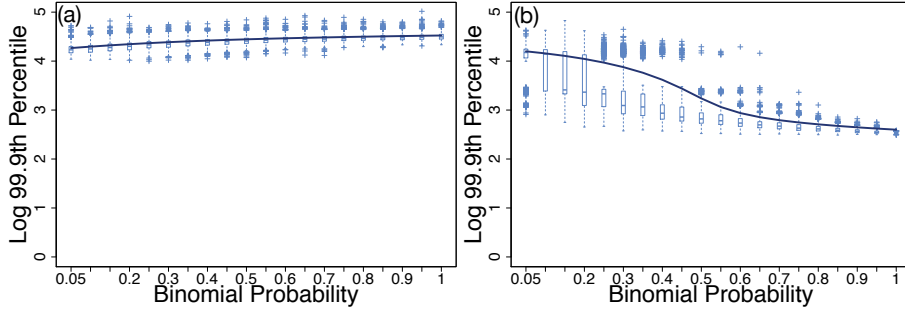
where  $\zeta(\kappa) := \sum_{k \in \mathbb{N}} a_k \kappa^k < \infty$  for some  $\kappa > 0$  and  $a_k \geq 0 \forall k \in \mathbb{N}$ . We denote this distribution by  $\text{Pow}(\kappa, \zeta)$  and the corresponding bounds on the tail probabilities of the steady-state waiting and response times in Theorem 3.3.1 evaluate to

$$\begin{aligned} P(W \geq \sigma) &\leq e^{\lambda\sigma} \frac{\kappa e^{-\mu\sigma} \zeta'(\kappa e^{-\mu\sigma})}{\zeta(\kappa)}, \\ P(R \geq \sigma) &\leq \frac{e^{\lambda\sigma}}{\rho} \frac{\kappa e^{-\mu\sigma}}{\zeta(\kappa)} [\kappa e^{-\mu\sigma} \zeta''(\kappa e^{-\mu\sigma}) + \zeta'(\kappa e^{-\mu\sigma})]. \end{aligned}$$

The derivation is provided in Appendix A. For a given form of  $\zeta$ , the strategy can be optimised to minimise the waiting times.

### 3.3.2 Homogeneous servers - partial scaling

In the previous section we considered linear scaling of the form (3.3.1) that models a perfect work division over  $l$  utilised servers in the sense of  $E[S_{n,i}] = E[S]/l$ . In this



**Figure 3.4:** The impact of the scheduling strategy on the waiting time percentiles. Simulation parameters:  $N = 10, \lambda = 0.9$ , parallelisation benefit: (a)  $\varphi = 0$  (b)  $\varphi = 0.2$ .

section, we analyse the general case of application specific scaling. Two prominent examples are: (i) MapReduce scenarios where the servers have to separately calculate a state before starting the task executions, and (ii) parallelised video transcoding, where some involved decoding operations have a diminishing return on parallelisation (Chong et al. 2007; Mesa et al. 2009). Mathematically, we assume that, for a certain application with scaling coefficient  $\varphi \in [0, 1]$ , the following scaling of service times holds,

$$S_{n,i} \mid \{L = l\} \stackrel{D}{=} \frac{S}{l^\varphi}. \quad (3.3.4)$$

Given  $\{L = l\}$ , the steady-state waiting times  $W$  and the response times  $R$  have the same representation as in (3.1.1) where we need to replace  $N$  with  $l$ . Now, we present our bounds in the partial scaling regime.

**Theorem 3.3.2.** *Consider a stable FJ system with  $N$  parallel work-conserving servers and renewal job arrivals with inter-arrival times  $A_i$ , for  $i \in \mathbb{N}$ . Let the random variable  $L \sim f_L \in \mathcal{P}([N], 2^{[N]})$  denote the number of servers chosen to split an incoming job. Let the unscaled service times  $S$  and the inter-arrival times  $T$  be exponentially distributed with parameters  $\mu$  and  $\lambda$ , respectively. For service times  $S_{n,i}$  at the  $n$ -th server that are scaled as in (3.3.4) for some  $\varphi \in [0, 1]$  the tail probabilities of the steady state waiting and response times are bounded as*

$$\begin{aligned} P(W \geq \sigma) &\leq e^{\lambda\sigma} E[L \exp(-\mu\sigma L^\varphi)], \\ P(R \geq \sigma) &\leq \frac{e^{\lambda\sigma}}{\rho} E[L^2 \exp(-\mu\sigma L^\varphi)], \end{aligned}$$

where  $\rho = \frac{\lambda}{\mu}$  is the unscaled utilisation level. The optimal strategy with respect to the bound for the waiting time is

$$L_{opt} \sim f_{opt} = \underset{f_L \in \mathcal{P}([N], 2^{[N]})}{\operatorname{argmin}} E[Le^{-\mu\sigma L^\varphi}].$$

The proof is provided in Appendix A. Remarkably, we find that for any fixed stochastic strategy, i.e.,  $p \in (0, 1]$  under no parallelisation benefit, the percentiles of the waiting

times grow as  $\mathcal{O}(\log \mathbb{E}[L])$ . In case of no stochastic scheduling, i.e.,  $p = 1$ , we recover the behaviour of  $\mathcal{O}(\log N)$  known from Baccelli, Makowski, and Shwartz (1989) and Rizk, Poloczek, and Ciucu (2015).

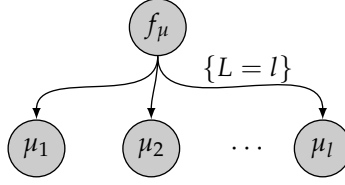
**Remark 3.3.2** (Insights into partial parallelisation benefit). Figure 3.3 conveys multiple insights into scheduling strategies under different application specific scaling  $\varphi$ . It depicts the mean waiting time in a given FJ system for different scheduling strategies given by the Binomial probability  $p$  for various parallelisation benefits given by the coefficient  $\varphi$ . The first insight from Figure 3.3 is the trade-off between the FJ inherent synchronisation penalty and the parallelisation benefit due to scaled service times. For a given scheduling strategy in an FJ system, i.e., the probability  $p$ , we observe a decrease in the mean waiting time with increasing scaling benefit  $\varphi$ . Second, for low parallelisation benefit  $\varphi$ , the synchronisation penalty predominates leading to an increase in mean waiting times. We note that this phenomenon also depends on the utilisation. Finally, for high parallelisation benefit  $\varphi$ , we observe a decay of the mean waiting times with  $p$ , i.e., essentially increasing the average number of utilised servers  $Np/(1 - q^N)$ . We observe a general diminishing behaviour with  $p$ . Hence, for larger  $\varphi$  substantial saving in server cost can be obtained by sacrificing a little in terms of the average waiting time. Figure 3.4 shows a similar behaviour for the percentiles of the waiting time distribution.

**OPTIMAL STRATEGY UNDER PARTIAL SCALING** The prime motive behind the analysis above is to gain analytic insights into the impact of the chosen number of servers on the waiting times for an application with a given scaling  $\varphi$  in a fixed FJ system. In particular, given a  $\varphi \in [0, 1]$ , we find the optimal stochastic scheduling strategy by minimising the bound obtained in Theorem 3.3.2. Observe that as  $\varphi \rightarrow 0$ , the scaling benefit diminishes to zero yielding the unscaled case from Section 3.1. Further, as  $\varphi \rightarrow 1$ , we get greater scaling benefit. The optimal strategy, therefore, would be to choose all the servers if the scaling benefit outweighs the synchronisation cost, and to choose only the strongest server if it does not. However, this depends on the parallelisation benefit  $\varphi$  specific to the given application.

### 3.3.3 Heterogeneous servers - a hierarchical model

In this section, we generalise our scaling discussion to the heterogeneous case, building on the analytic intuitions gained in the previous section. We argue that the average service times at different servers are not identical, but rather follow some suitable probability distribution (see Figure 3.5). We assume a randomly drawn server has an exponential service rate with parameter  $\mu$  where  $\mu$  itself is drawn from an underlying hierarchical distribution  $f_\mu$ . We present the following result for such a set-up, assuming the strict stability  $\max_{n \in [N]} \mathbb{E}[S_{n,1}] < \mathbb{E}[A_1]$ .

**Theorem 3.3.3.** Consider an FJ system with  $N$  parallel work-conserving servers fed by renewal job arrivals with iid exponentially distributed inter-arrival times  $A_i$  with parameter  $\lambda$ , for  $i \in \mathbb{N}$ . Let the random variable  $L \sim f_L \in \mathcal{P}([N], 2^{[N]})$  denote the number of servers chosen to split an



**Figure 3.5:** The hierarchical model for the heterogeneous FJ systems. Conditional on  $\{L = l\}$ , the average service rates are drawn from a hierarchical distribution  $f_\mu$ .

incoming job into and the unscaled service time  $S_n$  at the  $n$ -th server be exponentially distributed with parameter  $\mu_n \sim f_\mu$ . For service times  $S_{n,i}$  at the  $n$ -th server that are scaled as

$$S_{n,i} \mid \{L = l\} \stackrel{\mathcal{D}}{=} \frac{S_n}{l^\varphi},$$

independently for all  $n \in [l], i \in \mathbb{N}_0$ ,  $\varphi \in [0, 1]$ , the tail probabilities of the steady-state waiting and response times are bounded as

$$\begin{aligned} P(W \geq \sigma) &\leq e^{\lambda\sigma} \mathbb{E}[L \exp(-\min_{n \in [L]} \mu_n \sigma L^\varphi)], \\ P(R \geq \sigma) &\leq \frac{e^{\lambda\sigma}}{\lambda} \mathbb{E}[L^\varphi (\sum_{n \in [L]} \mu_n) \exp(-\min_{n \in [L]} \mu_n \sigma L^\varphi)]. \end{aligned}$$

The optimal strategy with respect to the bound above for the waiting time is given by

$$L_{opt} \sim f_{opt} = \underset{f_L \in \mathcal{P}([N], 2^{[N]})}{\operatorname{argmin}} \mathbb{E}[L \exp(-\min_{n \in [L]} \mu_n \sigma L^\varphi)].$$

The proof is provided in Appendix A.

**Example 3.3.1** (A two-class system). Consider the case where there are only two types of servers in the system, *fast* and *slow*. In a cloud computing infrastructure, these two types would correspond to different monetary prices. Suppose the exponential service rates of the two types of servers are  $\kappa_1$  and  $\kappa_2$ , respectively, and the arrival rate is  $\lambda$  with  $\lambda < \kappa_1 < \kappa_2$ . Denote the probability that a randomly drawn server is of type-1, i.e., has exponential service rate  $\kappa_1$ , by  $\pi$ . Hence, the service rate distribution is given by

$$f_\mu(x) := \pi \mathbb{1}(x=\kappa_1) (1 - \pi) \mathbb{1}(x=\kappa_2). \quad (3.3.5)$$

Given  $n$  random samples  $\mu_1, \mu_2, \dots, \mu_n$  from the above distribution, we require the first order statistic of the sample  $Y_n := \min_{i \in [n]} \mu_i$  to compute the bounds in Theorem 3.3.3. The distribution of  $Y_n$  is given by  $P(Y_n = \kappa_1) = 1 - (1 - \pi)^n = 1 - P(Y = \kappa_2)$ , such that its MGF is  $\mathbb{E}[\exp(aY_n)] = \exp(a\kappa_1) - (\exp(a\kappa_1) - \exp(a\kappa_2))(1 - \pi)^n$ , whence we can compute the bounds obtained in Theorem 3.3.3 for different choices of distributions of the number of used servers  $L$ . In particular, when  $L \sim \text{Binomial}(N, p)$  and we receive linear scaling  $\varphi = 1$ , the upper bounds on the tail probabilities can be written as

$$P(W \geq \sigma) \leq \exp(\lambda\sigma) \frac{Np}{1 - q^N} b_1(\sigma) \left[ 1 - (1 - \pi) \left( \frac{c_1(\sigma) - c_2(\sigma)}{b_1(\sigma)} \right) \right],$$

where, for  $i = 1, 2$ ,

$$\begin{aligned} b_i(\sigma) &:= \exp(-\sigma\kappa_i) (p \exp(-\sigma\kappa_i) + q)^{N-1}, \\ c_i(\sigma) &:= \exp(-\sigma\kappa_i) (p(1 - \pi) \exp(-\sigma\kappa_i) + q)^{N-1}. \end{aligned}$$

While the above example only considers two types of servers, it is worth mentioning that it can easily be extended to take into account finitely many types of servers.

**Example 3.3.2** (The hierarchical hyper-parameter model). In view of the stability of the system, we take  $f_\mu$  to be a truncated exponential with (hyper-) parameter  $\mu_0$ , truncated at  $\lambda$ . That is, we take

$$f_\mu(x) := \mu_0 \exp(-\mu_0(x - \lambda)) \mathbb{1}(x > \lambda). \quad (3.3.6)$$

Given  $n$  random samples  $\mu_1, \mu_2, \dots, \mu_n$  from the above distribution, the first order statistic of the sample  $Y_n := \min_{i \in [n]} \mu_i$  has a truncated exponential distribution with parameter  $n\mu_0$ , truncated at  $\lambda$ . The MGF of  $Y_n$  is given by  $E[\exp(aY_n)] = \exp(a\lambda) n\mu_0 / (n\mu_0 - a)$ . Taking the same approach as in Section 3.3.2, we can compute the waiting and response time bounds from Theorem 3.3.3 for different choices of distributions of  $L$ . In particular for the linear scaling case, *i.e.*,  $\varphi = 1$  and when  $L \sim \text{Binomial}(N, p)$ , the upper bounds on the tail probabilities can be explicitly found as (proof in Appendix A)

$$P(W \geq \sigma) \leq \frac{Np\mu_0}{(1 - q^N)(\mu_0 + \sigma)} (p \exp(-\sigma\lambda) + q)^{N-1}.$$

**Remark 3.3.3** (Heterogeneous FJ systems - three forces). As shown above the hierarchical model extends our findings in the previous sections to a wide setting providing insights and lending greater applicability. Theorem 3.3.3 shows that (i) the first order statistic  $Y_l := \min_{i \in [l]} \mu_i$  is decisive for the overall performance of the system, in addition, to the opposing forces from Section 3.3.1, *i.e.*, (ii) scaling of service times at each server due to the parallelisation, and (iii) the synchronisation penalty at the output. In fact, the heterogeneous case provides less scaling benefit than the homogeneous case due to  $Y_l$ . This impact can be directly seen from the position of  $Y_l$  in the exponent in Theorem 3.3.3. The optimal strategy given all the relevant parameter values is obtained, as before, by optimising the upper bound provided in Theorem 3.3.3.

In this chapter, we focussed on stochastic scheduling to decide the number of servers to choose in an FJ system. In Appendix A.4, we provide a numerical example highlighting a comparison between stochastic and deterministic strategies. In the next chapter, we shall discuss *provisioning*. We shall relax some of the technical assumptions, such as the independence of service and inter-arrival times, renewal arrival process etc. In particular, we shall derive an LDP for the steady-state waiting times under changing environments.



In order to model an FJ system, we assumed a renewal arrival process, iid service times in Chapter 3. However, recent evidences suggest that arrival processes such as the input to a MapReduce system or data centre traffic may not be renewal and may exhibit considerable burstiness (Y. Chen, Alspaugh, and Katz 2012; Heffes and Lucantoni 1986; Kandula et al. 2009; Yoshihara, Kasahara, and Takahashi 2001). Moreover, the servers may also be dependent in some sense, and may show phase-type behaviour. The behaviour of the inter-arrival times and the service times may change drastically depending on or being controlled by certain exogenous factors. For the purpose of mathematical abstraction, we use the term “environment” for these exogenous factors. In order to account for the effects of changing environment, we present a Markov-additive process (Iscoe, Ney, and Nummelin 1985) formulation (see Figure 4.1) in this chapter, and show how particular application scenarios can be derived as special cases of this formulation. In particular, we cover three application scenarios: (i) non-renewal (Markov-modulated) arrivals, (ii) servers showing phase-type behaviour, and (iii) Markov-modulated arrivals and service. Finally, we bring in a notion of provisioning, an umbrella term used for a rule that decides on the FJ job division into tasks, or that regulates service rates either *reactively* or *proactively*. Proactive provisions anticipate the change of environment, and act accordingly, while reactive provisions only *react* to the current environment.

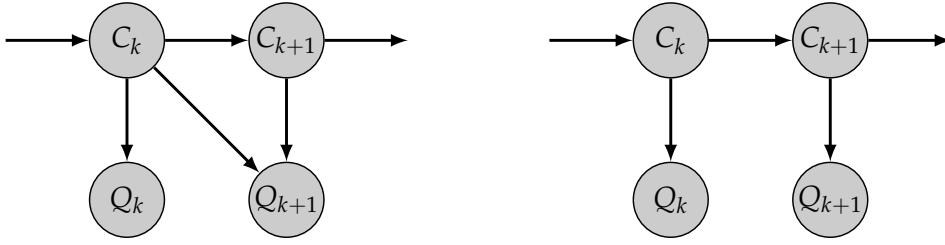
#### 4.1 MARKOV-ADDITIVE PROCESS FORMULATION

The roadmap is as follows: we first establish an LDP for an FJ system based on a Markov-additive process representation. Based on the LDP, we provide computable bounds on the tail probabilities of the steady-state waiting times. The idea is to use these general results to obtain several special cases that are relevant for practical purposes.

##### 4.1.1 System description

Consider the single stage FJ queueing system with  $N$  parallel servers as discussed in Section 2.2 (also see Figure 2.1). Jobs arrive at the input station according to some process with inter-arrival time  $A_i$  between the  $i$ -th and  $(i + 1)$ -th job,  $i \in \mathbb{N}$ . A job is split into  $N$  tasks each of which is assigned to exactly one server. The service time for the task of job  $i$  at the  $n$ -th server is denoted by the random variable  $S_{n,i}$ , where  $n \in [N]$  (see Figure 2.1). Finally the job leaves the system when *all* of its tasks are served, posing a synchronisation constraint at the output. We assume the servers are work-conserving.

In real applications, the behaviour of the inter-arrival times and the service times may change drastically depending on certain exogenous factors. For example, during a heavy traffic period, the inter-arrival times are much shorter compared to those during a low traffic period. From considerations of energy conservation or cost, the service times may also be modulated externally to yield high or low efficiency. For instance,



**Figure 4.1:** Graphical representation of a Markov-additive process  $\{C_k, Q_k\}_{k \in \mathbb{N}}$  (**Left**) and its special “uncoupled” case, the Markov-modulated process (**Right**). The nodes represent the variables and the arrows, the dependence structure. Please note that  $Q_k$  is an additive component, *i.e.*,  $Q_{k+1} = Q_k + (X_{1,k+1}^A, X_{2,k+1}^A, \dots, X_{N,k+1}^A)$ . While the Markov-additive process, from the perspective of provisioning, is capable of modelling “proactive” systems (that anticipate the immediate future and act accordingly, *e.g.*, set service rates proactively) as well as reactive systems (that react on the current environment), the uncoupled process on the right is only capable of modelling the latter.

given a fixed monetary budget, the service rates of a cloud computing service such as the Amazon AWS (Amazon.com, Inc. 2017) could be altered as the price changes to meet the budget constraint. For the purpose of mathematical abstraction, we use an umbrella term “environment” for these exogenous factors. In order to capture the effects of changing environment, we consider an underlying Markov chain  $\{C_k\}_{k \in \mathbb{N}_0}$  on some measurable space  $(\mathbb{E}, \mathcal{E})$ . Note that  $\mathbb{E}$  need not be finite, or even countable. The Markov chain can be used to capture the changes in job arrival rates, *i.e.*, modulate the arrival process; to decide the service rates of the servers, *i.e.*, modulate the service process; or both in which case it is said to modulate both the arrival as well as the service processes. Naturally, different choices of the state space  $\mathbb{E}$  yield different types of modulation to suit different real-life applications. In Table 4.1, we present a glossary of examples of  $\mathbb{E}$ . Detailed examples will be provided in later sections.

#### 4.1.1.1 Waiting times

Recall the difference process  $Q_k$  on  $(\mathbb{R}^N, \mathcal{B}(\mathbb{R}^N))$  defined in Section 2.2 as follows

$$Q_k := (X_{1,k}, X_{2,k}, \dots, X_{N,k}) \text{ with } X_{n,k} := \sum_{i=1}^k X_{n,i}^A, \quad (4.1.1)$$

where  $X_{n,i}^A = S_{n,i} - A_i$  for all  $i \in \mathbb{N}$  and set  $X_{n,0} := 0$ , for each  $n \in [N]$ . Then, we have the following steady-state representation for the waiting time  $W$

$$W \stackrel{\mathcal{D}}{=} \max_{k \in \mathbb{N}_0} \max_{n \in [N]} X_{n,k}. \quad (4.1.2)$$

In the next section, we establish an LDP for the waiting times under mild assumptions following Duffield (1994), Iscoe, Ney, and Nummelin (1985), and Ney and Nummelin (1987).

Modulation	State space	Scenario
Only arrivals	$\mathbb{E} = \{0, 1\}$ $\mathbb{E} = \{1, 2, \dots, d\}$ $\mathbb{E} = \mathbb{N}$ $\mathbb{E} = \mathbb{E}^A \subseteq \mathbb{R}$	Markov-modulated high-low (or on-off) arrivals. Finite state modulation of the arrivals. Countable state modulation of the arrivals. Modulation of the arrivals on an uncountable state space. <i>Real-life example:</i> bursty input at MapReduce clusters.
Only service	$\mathbb{E} = \{0, 1\}$ $\mathbb{E} = \{1, 2, \dots, d\}$ $\mathbb{E} = \{0, 1\} \times \{0, 1\} \times \dots \times \{0, 1\}$ $\mathbb{E} = \{0, 1\} \times \dots \times \{0, 1\} \times \{1, 2, \dots, d\}$ $\mathbb{E} = \{1\} \times \{1\} \times \dots \times \{0, 1\}$ $\mathbb{E} = \mathbb{E}_1^S \times \mathbb{E}_2^S \times \dots \times \mathbb{E}_N^S$	All servers are Markov high-low modulated. All servers are Markov modulated on a finite set. All servers are Markov high-low modulated, but by separate chains that may or may not be independent. All but the $N$ -th server are Markov high-low modulated by separate chains and the $N$ -th server is Markov modulated on a finite set. Only the $N$ -th server is Markov high-low modulated. The $n$ -th server is modulated on its own state space $\mathbb{E}_n^S \subseteq \mathbb{R}$ , for $n \in [N]$ . <i>Real-life example:</i> switching between cloud service machines such as Amazon AWS under a monetary budget constraint, as the prices change over time; provisioning such as round-robin in MapReduce clusters.
Both arrivals and service	$\mathbb{E} = \mathbb{E}^A \times \mathbb{E}_1^S \times \mathbb{E}_2^S \times \dots \times \mathbb{E}_N^S$	The arrival process is modulated on state space $\mathbb{E}^A \subseteq \mathbb{R}$ and the $n$ -th server is modulated on its own state space $\mathbb{E}_n^S \subseteq \mathbb{R}$ , for $n \in [N]$ . The modulating chains need not be independent. <i>Real-life example:</i> adaptive provisioning (both proactive and reactive) in parallel systems such as MapReduce clusters; modulation in Multi-path TCP.
No modulation	$\mathbb{E} = \{1\}$	Reduces to the renewal case.

**Table 4.1:** Table showing different choices for the state space for different application scenarios.

#### 4.1.2 Large deviations of the waiting times

We assume that the process  $\{(C_k, Q_k)\}_{k \in \mathbb{N}_0}$  is a Markov-additive process on  $(\mathbb{E} \times \mathbb{R}^N, \mathcal{E} \times \mathcal{B}(\mathbb{R}^N))$ . To be precise,

**Definition 4.1.1. (Markov-additive process)** The processes  $\{(C_k, Q_k)\}_{k \in \mathbb{N}}$  is a Markov-additive process on  $(\mathbb{E} \times \mathbb{R}^N, \mathcal{E} \times \mathcal{B}(\mathbb{R}^N))$  if

1. The process  $\{(C_k, Q_k)\}_{k \in \mathbb{N}}$  is a Markov process on  $(\mathbb{E} \times \mathbb{R}^N, \mathcal{E} \times \mathcal{B}(\mathbb{R}^N))$ .
2. The following holds for  $c \in \mathbb{E}, s \in \mathbb{R}^N, F \in \mathcal{E}, G \in \mathcal{B}(\mathbb{R}^N)$ ,

$$\begin{aligned} & P((C_{k+1}, Q_{k+1}) \in F \times (G + s) \mid (C_1, Q_1) = (c, s)) \\ &= P((C_{k+1}, Q_{k+1}) \in F \times G \mid (C_1, Q_1) = (c, 0)) \\ &= P((C_{k+1}, Q_{k+1}) \in F \times G \mid C_1 = c). \end{aligned}$$

The Markov chain  $C_k$  is endowed with an additive component  $Q_k$ , the difference process in our queueing system defined in (4.1.1). Note that the difference process  $Q_k$  is indeed additive in the sense that  $Q_{k+1} = Q_k + (X_{1,k+1}^A, X_{2,k+1}^A, \dots, X_{N,k+1}^A)$ . Intuitively, the environment captured by the Markov chain  $C_k$  modulates the inter-arrival and service times (through their difference) not only for the current job but also for the next arriving job (see Figure 4.1). Accordingly, define the transition kernel

$$L(c, F \times G) := P((C_1, Q_1) \in F \times G \mid Q_0 = c), \quad (4.1.3)$$

where  $c \in \mathbb{E}, F \in \mathcal{E}$ , and  $G \in \mathcal{B}(\mathbb{R}^N)$ . We need the following additional technical assumptions, such as uniform recurrence of the Markov chain, stability of the queueing system, and moment conditions.

**A1 (Recurrence)** The process  $\{C_k\}_{k \in \mathbb{N}_0}$  is an aperiodic, irreducible Markov chain with respect to some maximal irreducibility measure and there exists a probability measure  $\nu$  on  $(\mathbb{E} \times \mathbb{R}^N, \mathcal{E} \times \mathcal{B}(\mathbb{R}^N))$ , an integer  $m$ , and real numbers  $0 < b_0 \leq b_1 < \infty$  such that

$$b_0 \nu(F \times G) \leq L^m(x, F \times G) \leq b_1 \nu(F \times G),$$

where  $L^m(x, F \times G) := P((C_m, Q_m) \in F \times G \mid C_1 = x)$ , for each  $x \in \mathbb{E}, F \in \mathcal{E}$  and  $G \in \mathcal{B}(\mathbb{R}^N)$ .

**A2 (Exponential transform)** Consider the exponential transform of  $\nu$ ,

$$\tilde{\nu}(F, s) := \int_{\mathbb{R}^N} \nu(F \times dy) \exp(sy). \quad (4.1.4)$$

We assume that  $\mathcal{D}_0 := \mathcal{D}\tilde{\nu}(\mathbb{E}, \cdot)$  is open, treating  $\tilde{\nu}(\mathbb{E}, \cdot)$  as a function on  $\mathbb{R}^N$ . The openness renders analyticity and essential smoothness to the logarithm of the maximal, simple eigenvalue of the transformed kernel  $\tilde{L}$  in (4.1.5). This will be clear in the proof of Theorem 4.1.1.

**A3 (Stability)** For stability of the queueing system, we assume  $\max_{n \in [N]} E[X_{n,1}] < 0$ .

**A4 (Cumulants)** Allowing possibly infinite values, define, for  $s \in \mathbb{R}, n \in [N]$ ,

$$\begin{aligned} \lambda_k^{(n)}(s) &:= k^{-1} \log E[\exp(sX_{n,k})], \\ \lambda^{(n)}(s) &:= \lim_{k \rightarrow \infty} k^{-1} \log E[\exp(sX_{n,k})]. \end{aligned}$$

To exclude pathological cases, we assume that the effective domains of  $\lambda_k^{(n)}$  and  $\lambda^{(n)}$  include common open interval containing 0. This moment condition is required for the establishment of an LDP.

Exponential transforms play a vital role in the study of large deviations (Dembo and Zeitouni 2010; Varadhan 2016). In fact, the exponential transform of the transition kernel together with its largest eigenvalue eventually yield an LDP (Iscio, Ney, and Nummelin 1985). Therefore, define the following exponential transform of the transition kernel defined in (4.1.3), for all  $c \in \mathbb{E}$ ,  $F \in \mathcal{E}$ , and  $s \in \mathbb{R}^N$ ,

$$\tilde{L}(c, F; s) := \int_{\mathbb{R}^N} L(c, F \times dy) \exp(sy). \quad (4.1.5)$$

Our strategy is to first establish an LDP for  $\{(C_k, Q_k)\}_{k \in \mathbb{N}_0}$  making use of standard results from probability theory and then, use that to arrive at an LDP for the waiting times in the queueing system via the contraction principle of large deviations theory (Dembo and Zeitouni 2010). Before presenting our result, we introduce the following notation that we make use of while applying the contraction principle. For  $y \in \mathbb{R}$ , define

$$Y_N(y) := \bigcup_{F \in \{S \subseteq [N] : S \neq \emptyset\}} G_F, \quad (4.1.6)$$

where

$$G_F := B_1 \times B_2 \times \dots \times B_N \text{ such that } B_i = \begin{cases} \{y\} & \text{if } i \in F, \\ \mathbb{R} \setminus [y, \infty) & \text{if } i \in [N] \setminus F \end{cases}.$$

The set  $Y_N(y)$  is the union of all  $N$ -fold Cartesian products of sets at least one of which is  $\{y\}$  and all others are  $(-\infty, y)$ . For example,

$$Y_2(y) = \{y\} \times (-\infty, y) \bigcup (-\infty, y) \times \{y\} \bigcup (y, y).$$

Note that, for each  $y \in \mathbb{R}$ , the set  $Y_N(y)$  is a Borel set.

**Theorem 4.1.1** (Large deviations principle). *Assume A1, A2, A3, and A4. Then, for each  $\theta \in \mathcal{D}_0$  defined in A3, the transformed kernel  $\tilde{L}$  in (4.1.5) has a maximal, real, simple eigenvalue  $\lambda(\theta)$ . Moreover, the waiting times  $W_k$  satisfy a large deviations principle with a good rate function  $J : \mathbb{R} \rightarrow \mathbb{R}$ ,*

$$\limsup_{k \rightarrow \infty} k^{-1} \log \mathbb{P}(W_k \in B) \leq - \inf_{y \in \text{Cl } B} J(y) \quad (4.1.7)$$

$$\liminf_{k \rightarrow \infty} k^{-1} \log \mathbb{P}(W_k \in B) \geq - \inf_{y \in \text{Int } B} J(y), \quad (4.1.8)$$

for all  $B \in \mathcal{B}(\mathbb{R})$ , where

$$J(y) := \inf_{x \in Y_N(y)} \Lambda^*(x), \quad (4.1.9)$$

$$\Lambda^*(x) := \sup_{z \in \mathbb{R}^N} \{zx - \log \lambda(z)\}. \quad (4.1.10)$$

The proof of Theorem 4.1.1 follows by first establishing an LDP for  $\{(C_k, Q_k)\}_{k \in \mathbb{N}_0}$  using Iscoe, Ney, and Nummelin (1985) and Ney and Nummelin (1987) and then applying the contraction principle (Dembo and Zeitouni 2010). For the sake of completeness we provide it in Appendix B.1. The Theorem 4.1.1 provides estimates of probabilities of rare events such as the waiting times making large deviations from its mean value. Moreover, the rate function  $J$  is unique and therefore, uniquely characterises the asymptotic behaviour of the waiting times (Dembo and Zeitouni 2010; Ganesh, O’Connell, and Wischik 2004; Varadhan 2016). It is remarkable that it is possible to estimate probabilities of rare events under mild technical conditions A1, A2, A3 and A4. For practical purposes, however, the computation of the rate function  $J$  involves the joint distribution of  $Q_k$ , which, in turn, involves the joint distribution of the inter-arrival times and the service times at different servers. This computation may not be easy to perform for arbitrary choices of probability distributions of the inter-arrival times and the services times. Therefore, in the next section, we make a few simplifying assumptions for the sake of computability, and provide a computable upper bound on the tail probabilities of the steady-state waiting times. The bound is derived as a by-product of the large deviations result.

#### 4.1.3 Simplifications for computability: probabilistic bounds on waiting times

In addition to A1, A2, A3 and A4, we assume that conditional on  $\{C_k = c\}$ , the servers act independently. This entails that the processes  $\{(C_k, X_{n,k})\}_{k \in \mathbb{N}}$ , for each  $n \in [N]$  are Markov-additive processes on  $(\mathbb{E} \times \mathbb{R}, \mathcal{E} \times \mathcal{B}(\mathbb{R}))$ . Their transition kernels are defined as, for  $n \in [N]$ ,

$$K_n(c, F \times G) := P((C_1, X_{n,1}) \in F \times G \mid C_0 = c), \quad (4.1.11)$$

where  $c \in \mathbb{E}$ ,  $F \in \mathcal{E}$  and  $G \in \mathcal{B}(\mathbb{R})$ . Please note the difference to (4.1.3). Also, define the corresponding exponential transforms

$$\tilde{K}_n(c, F; s) := \int_{\mathbb{R}} K_n(c, F \times dx) \exp(sx), \quad \forall n \in [N]. \quad (4.1.12)$$

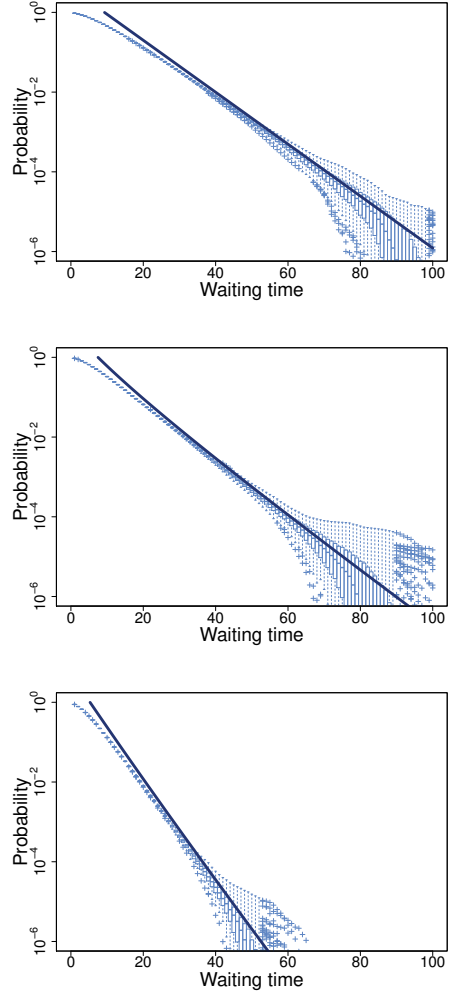
We proceed to construct martingales using the largest eigenvalues of the transformed kernels, and then apply the celebrated Doob’s martingale inequality (Durrett 2010a) on each of  $X_{n,k}$  for  $n \in [N]$ . This step essentially yields bounds on server-specific waiting times. Coupled with the assumption of conditional independence of the servers, we obtain an upper bound on the tail probability of the steady-state waiting time of the entire queueing system. These ideas are made precise in the proof of the following theorem providing upper bound on the tail probability of the steady-state waiting times in an FJ system with  $N$  heterogeneous work-conserving servers.

**Theorem 4.1.2** (Work-conserving systems). *Consider an FJ system with  $N$  parallel work-conserving servers, as described in Section 4.1.1. Then, we have*

1. For all  $n \in [N]$  and  $s \in \mathcal{D}\lambda^{(n)}$ ,  $\exp(\lambda^{(n)}(s))$  is the simple maximal eigenvalue of  $\tilde{K}_n$ , and the corresponding right eigenfunction  $\{r_n(c, s); c \in \mathbb{E}\}$  satisfying

$$\exp(\lambda^{(n)}(s))r_n(c, s) = \int_{\mathbb{R}} \tilde{K}_n(c, d\tau; s)r_n(\tau, s),$$

**Figure 4.2:** Numerical verification of the bounds (solid) vs. simulation box plots for work-conserving systems. **(Top)** An FJ system with Markov-modulated arrivals. The modulating Markov chain takes values in the set  $\mathbb{E} = \{1, 2, 3, 4\}$ . The exponential inter-arrival times have parameters 0.70, 0.75, 0.90, and 0.95. **(Middle)** An FJ system with Markov-modulated service times. The modulating Markov chain takes values in the set  $\mathbb{E} = \{1, 2, 3, \dots, 32\}$ . The exponential inter-arrival times have parameter 0.9. **(Bottom)** An FJ system with Markov-modulated arrival and service times. The modulating Markov chain takes values in the set  $\mathbb{E} = \{1, 2, 3, \dots, 64\}$ . In all the cases, there are five heterogeneous and work-conserving servers whose service rates are drawn randomly, satisfying the stability conditions in A3 and A4. The transition probabilities and the initial distribution of  $C_k$  are chosen randomly. Observe that the analytic bounds obtained in Theorem 4.1.2 are in close agreement with the sample estimates of the tail probabilities of the waiting times.



is positive and bounded above.

2. The tail probabilities of the steady-state waiting times are bounded above by

$$P(W \geq w) \leq \sum_{n \in [N]} \phi_n(\theta_n) \exp(-\theta_n w), \quad (4.1.13)$$

where  $\theta_n := \sup\{s > 0 \mid \lambda^{(n)}(s) \leq 0\}$  and  $\phi_n(s) := \text{ess sup}\{\mathbb{1}(X_{n,1} > 0)/r_n(C_1, s)\}$ , after having normalised  $r_n(\cdot, \theta_n)$  so that  $E[r_n(C_0, \theta_n)] = 1$ , for each  $n \in [N]$ .

The existence of the simple maximal eigenvalue is guaranteed by Harris (1963, Chapter III, Theorem 10.1). The proof of Theorem 4.1.2 follows by extending results for Markov-additive processes from probability literature (see, e.g., Iscoe, Ney, and Nummelin (1985, Lemma 3.1 and 3.2) and also Duffield (1994)). However, for the sake of completeness, it is provided in Appendix B.2. This theorem is central to all the application scenarios that we consider in this chapter. The quantity  $\theta_n$  is called the decay rate

of the  $n$ -th server, and the quantity  $\tilde{\theta} := \min_{n \in [N]} \theta_n$  is defined to be the decay rate of the system. The latter definition is motivated from the principle of largest exponent in large deviations theory (Dembo and Zeitouni 2010, Lemma 1.2.15), which roughly states that, on an exponential scale, the effective rate of a sum of finitely many sequences is governed by the maximum of them. This corroborates the intuition that the system is constrained by the weakest (slowest) of the servers. The quantities  $\phi_n$ 's are called prefactors.

The bound provided in Theorem 4.1.2 is computable. An interesting observation is that, given the transition kernel  $T$  of the Markov chain  $C_k$  alone, one can view the transformation defined in (4.1.12) as a transformation of  $T$  also. This point of view is useful for computational purposes. In the following, we provide two illustrations.

**Example 4.1.1.** Suppose there are two heterogeneous servers labelled 1 and 2. We are interested in modelling two different environments, *i.e.*, we set  $\mathbb{E} = \{1, 2\}$ . In keeping with Figure 4.1, we assume the inter-arrival times and the services times at the  $n$ -th server are exponentially distributed with rates  $\lambda_{i,j}$  and  $\mu_{i,j}^{(n)}$  respectively, when the underlying Markov chain  $C_k$  transitions from state  $i$  to state  $j$ , for  $i, j, n = 1, 2$ . The  $\lambda$ 's and the  $\mu$ 's are taken to be strictly positive to avoid trivialities. Assume the inter-arrival times and the service times are independent, conditional on the Markov chain. Let  $T := ((t_{i,j}))_{i,j=1,2}$  denote the transition probability matrix of the Markov chain  $C_k$ . Then, for  $n = 1, 2$ , the random variable  $X_{n,1}$  is a difference of two exponential random variables, and therefore,  $K_n(c_i, \{c_j\} \times B) = t_{i,j} \int_B f_n(y) dy$  (see Figure 4.1), where

$$f_n(y) := \begin{cases} \left( \frac{1}{\mu_{i,j}^{(n)}} + \frac{1}{\lambda_{i,j}} \right)^{-1} \exp(\lambda_{i,j} y) & \text{if } y \leq 0 \\ \left( \frac{1}{\mu_{i,j}^{(n)}} + \frac{1}{\lambda_{i,j}} \right)^{-1} \exp(-\mu_{i,j}^{(n)} y) & \text{if } y > 0. \end{cases}$$

The transformed kernel is the conditional MGF of the random variable  $X_{n,1}$ . The exponentially transformed kernels are  $\tilde{K}_n(c_i, \{c_j\}; s) = t_{i,j} \left( \frac{\mu_{i,j}^{(n)}}{\mu_{i,j}^{(n)} - s} \right) \left( \frac{\lambda_{i,j}}{\lambda_{i,j} + s} \right)$ . The decay rates  $\theta_n$ 's are obtained by computing the largest eigenvalues of the transformed matrices

$$\begin{pmatrix} t_{1,1} & t_{1,2} \\ t_{2,1} & t_{2,2} \end{pmatrix} \mapsto \begin{pmatrix} t_{1,1} \left( \frac{\mu_{1,1}^{(n)}}{\mu_{1,1}^{(n)} - s} \right) \left( \frac{\lambda_{1,1}}{\lambda_{1,1} + s} \right) & t_{1,2} \left( \frac{\mu_{1,2}^{(n)}}{\mu_{1,2}^{(n)} - s} \right) \left( \frac{\lambda_{1,2}}{\lambda_{1,2} + s} \right) \\ t_{2,1} \left( \frac{\mu_{2,1}^{(n)}}{\mu_{2,1}^{(n)} - s} \right) \left( \frac{\lambda_{2,1}}{\lambda_{2,1} + s} \right) & t_{2,2} \left( \frac{\mu_{2,2}^{(n)}}{\mu_{2,2}^{(n)} - s} \right) \left( \frac{\lambda_{2,2}}{\lambda_{2,2} + s} \right) \end{pmatrix} \text{ for } n = 1, 2.$$

Let  $r_1$  and  $r_2$  denote the corresponding right eigenvectors after having carried out the normalisation to get  $E[r_1(C_0, \theta_1)] = 1$ , and  $E[r_2(C_0, \theta_2)] = 1$ . Denoting the initial distribution of the chain  $C_k$  by  $\pi = (\pi_1, \pi_2)$ , the normalization amounts to setting  $r_1(1, \theta_1)\pi_1 + r_1(2, \theta_1)\pi_2 = 1$  and  $r_2(1, \theta_2)\pi_1 + r_2(2, \theta_2)\pi_2 = 1$ . Because of the exponential assumption, the prefactors are given by  $\phi_1(\theta_1) = \max(1/r_1(1, \theta_1), 1/r_1(2, \theta_1))$ , and  $\phi_2(\theta_2) = \max(1/r_2(1, \theta_2), 1/r_2(2, \theta_2))$ . Finally, following (4.1.13), we get the bound

$$P(W \geq w) \leq \phi_1(\theta_1) \exp(-\theta_1 w) + \phi_2(\theta_2) \exp(-\theta_2 w).$$



**Example 4.1.2.** Similar to Example 4.1.1, let us assume there are two heterogeneous servers labelled 1 and 2. However, in contrast to Example 4.1.1, we assume the Markov chain has an uncountable state space, e.g., an interval  $[a, b]$ . Conforming to the dependence structure dictated by the graphical model shown in Figure 4.1, we assume the inter-arrival times and the services times at the  $n$ -th server are exponentially distributed with strictly positive rate functions  $\lambda(x, y)$  and  $\mu^{(n)}(x, y)$  respectively, when the underlying Markov chain  $C_k$  transitions from state  $x$  to state  $y$ , for  $n = 1, 2$  and  $x, y \in [a, b]$ . For simplicity, we also assume the inter-arrival times and the service times are independent, conditional on the Markov chain. The transition kernel of  $C_k$  is denoted by  $T$ , as before. The choices of the rate functions  $\lambda$  and  $\mu^{(n)}$ , and the transition kernel  $T$  depend on the specific application scenario. For instance, if the environment in question does not vary drastically for two consecutive incoming jobs, we may choose a Gaussian kernel with a small variance or a Laplace kernel with a small scale parameter, both restricted to  $[a, b]$ . We can control how rapidly the environment changes via the variance parameter of the Gaussian kernel or the scale parameter of the Laplace kernel. In this example, let us take  $T$  to be the Laplace kernel with scale parameter  $\sigma$ . Then, doing similar calculation as in Example 4.1.1, we get

$$\tilde{K}_n(x, F; s) = \frac{1}{u(x)} \int_F \exp\left(-\frac{|y-x|}{\sigma}\right) \left( \frac{\mu^{(n)}(x, y)}{\mu^{(n)}(x, y) - s} \right) \left( \frac{\lambda(x, y)}{\lambda(x, y) + s} \right) dy,$$

where  $u(x) = \int_a^b \exp\left(-\frac{|y-x|}{\sigma}\right) dy$ , and  $x \in [a, b]$ . Given the choices of the rate functions  $\lambda$  and  $\mu^{(n)}$ , we find the maximal eigenvalue and the corresponding right eigenfunction of  $\tilde{K}_n$  to obtain the bound given in (4.1.13). The eigenvalue and the right eigenfunction are usually found as a solution to the integral equation mentioned in Theorem 4.1.2. Note that finding closed-form expressions may be infeasible for arbitrary choices of the rate functions  $\lambda$  and  $\mu^{(n)}$ . In such a situation, we resort to numerical methods (Atkinson 2008; Rasmussen and Williams 2006). A standard approach is to approximate the integral using samples (Baker 1977).

For the sake of simplicity, let us assume that the environment only modulates the arrival process. In particular, when the Markov chain is in state  $x$ , the inter-arrival times are assumed to be exponentially distributed with rate  $x$ , i.e.,  $\lambda(x, y) = x$ . The task of finding the maximal eigenvalue of the transformed kernel  $\tilde{K}_n$  is equivalent to solving the following integral equation for  $\lambda^{(n)}$ , and  $r_n$ ,

$$\int_a^b \exp\left(-\frac{|x-y|}{\sigma}\right) r_n(x, s) dx = U_n(y, s) \exp(\lambda^{(n)}(s)) r_n(y, s),$$

where the conditional MGF accounting for the service process as well as the constants have been absorbed into the function  $U_n(y, s) = \left(1 + \frac{s}{y}\right) \left(1 - \frac{s}{\mu^{(n)}}\right) u(y)$ . In order to solve the above integral equation, we differentiate it twice with respect to  $y$  to obtain the following differential equation,

$$r_n''(y, s) + 2 \frac{U_n'(y, s)}{U_n(y, s)} r_n'(y, s) + \left( \frac{U_n''(y, s)}{U_n(y, s)} - \frac{1}{\sigma^2} \left( 1 - \frac{2\sigma \exp(-\lambda^{(n)}(s))}{U_n(y, s)} \right) \right) r_n(y, s) = 0. \quad (4.1.14)$$

The derivation of (4.1.14) is provided in Appendix B.2. The nonlinear differential equation (4.1.14) can then be solved numerically. After doing necessary normalisation to get  $E[r_n(C_0, \theta_n)] = 1$ , for  $n = 1, 2$ , we obtain the bound using Theorem 4.1.2.

For ease of computation, in the following we shall consider what is referred to as the “uncoupled” MA process in Iscoe, Ney, and Nummelin (1985). This essentially refers to a process with Markov-modulated increments (see Figure 4.1 and refer to Duffield (1994)). This is an important class from a practical perspective, specially in the light of recent empirical evidences of burstiness in clusters running MapReduce (Y. Chen, Alspaugh, and Katz 2012; Heffes and Lucantoni 1986; Kandula et al. 2009; Yoshihara, Kasahara, and Takahashi 2001).

#### 4.1.4 The “uncoupled” case

Suppose the distributions of increments,  $X_{n,k+1}^A$ , for each  $n \in [N]$ , do not depend on  $C_k$ , conditional on  $C_{k+1}$  (see Figure 4.1). This allows us to find conditional distributions  $Q_n(c, B) := P(X_{n,1}^A \in B \mid C_1 = c)$ , for each  $n \in [N]$  and for each  $c \in \mathbb{E}$  and  $B \in \mathcal{B}(\mathbb{R})$ . Then, the transformed kernels in (4.1.12) simplify as follows

$$\tilde{K}_n(c, d\tau; s) = T(c, d\tau) \int_{\mathbb{R}} Q_n(\tau, dz) \exp(sz) = T(c, d\tau) E_{\tau} \left( \exp(sX_{n,1}^A) \right).$$

Here we use the shorthand notation  $E_{\tau} \left( \exp(sX_{n,1}^A) \right)$  to denote  $E[\exp(sX_{n,1}^A) \mid C_1 = \tau]$ , the MGF of  $X_{n,1}^A$  conditioned on  $\{C_1 = \tau\}$ , the event that underlying Markov chain is in state  $\tau \in \mathbb{E}$  for the first arrival. We can further simplify the formulas if we make following assumptions<sup>1</sup>.

**U1** We assume that the service times and the arrival times are independent, conditioned on  $\{C_k = c\}$ . This yields

$$\tilde{K}_n(c, d\tau; s) = T(c, d\tau) E_{\tau} \left( \exp(sS_{n,1}) \right) E_{\tau} \left( \exp(-sA_1) \right). \quad (4.1.15)$$

**U2** If further the increments  $X_{n,1}^A$  take positive values with non-zero probability for any conditioning of  $C_k$ , then the essential supremums in Theorem 4.1.2 simplify to

$$\phi_n(s) = \sup_{c \in \mathbb{E}} \{1/r_n(c, s)\}. \quad (4.1.16)$$

With these simplifications the computation of the bound on the tail probabilities of the waiting times is easier. We present the procedure in the form of Algorithm 4.1 for ease of understanding and implementation. Note that Algorithm 4.1 requires numerical solution methods when closed-form analytic expressions are difficult to obtain.

<sup>1</sup> These assumptions are made only for the sake of simplification of computation, and are not necessary for the bounds of the general case.

**Algorithm 4.1** Pseudocode for work-conserving systems**Require:** Transition kernel  $T$ , and the MGFs  $E_\tau(\exp(sS_{n,1}))$ ,  $E_\tau(\exp(-sA_1))$ 


---

```

1: if A1 and A2 and A3 and A4 then
2:   for  $n \in [N]$  do
3:     Transform  $T$  to get  $\tilde{K}_n(c, d\tau; s)$  (see (4.1.15))
4:      $\exp(\lambda^{(n)}(s)) \leftarrow$  maximal eigenvalue of  $\tilde{K}_n(c, d\tau; s)$ 
5:      $\theta_n \leftarrow \sup\{s > 0 \mid \lambda^{(n)}(s) \leq 0\}$ 
6:     Normalise  $r_n(\cdot, \theta_n)$  so that  $E[r_n(C_0, \theta_n)] = 1$ 
7:      $\phi_n(\theta_n) \leftarrow \sup_{c \in \mathbb{E}} \{1/r_n(c, \theta_n)\}$ 
8:   end for
9: end if
10: return The decay rates  $\theta_n$  and the prefactors  $\phi_n$ 

```

---

**Remark 4.1.1** (Bounds on the mean waiting times). The bound in (4.1.13) can also be used to derive an upper bound on the mean waiting time for the work-conserving system as follows

$$E[W] \leq \sum_{n \in [N]} \frac{\phi_n(\theta_n)}{\theta_n}. \quad (4.1.17)$$

So far we have considered only work-conserving servers. However, there are situations when the assumption of work-conservingness is not tenable. In particular, there are many real-life application scenarios where the servers are blocking in nature. Such a server waits for all other servers to finish servicing the tasks of the current job before taking up the next job. This entails forced idleness resulting in higher waiting times. In the next section, we show that our framework, although designed for work-conserving systems, is applicable to blocking systems as well and yields computable probability bounds by treating an FJ system with  $N$  blocking servers as a virtual queueing system with just one server. In that sense, blocking FJ systems can also be analysed within our framework as a special case.

## 4.2 BLOCKING SYSTEMS

Blocking systems arise naturally in several real-life applications, for instance, when the dispatcher and the task collector in Figure 2.1 are one and the same unit that assigns new jobs only after the current job is executed. In a parallel computation scenario, the master node, upon arrival of a computation request, may assign intermediate tasks to a number of slave nodes, then wait for all the slave nodes to hand over their intermediate results back to the master node for further aggregation before assigning new computation tasks to the slave nodes. Blocking systems also arise when there needs to be a consensus among the servers regarding the job division (with respect to fairness or some other criterion) before its tasks can be executed.

There is an additional layer of synchronisation in a blocking FJ system. All the servers start servicing the tasks of a job at the same time. Servers that are finished executing

the task of the current job wait for all other servers to finish theirs before taking up the task of the next job. Therefore, the waiting time  $W'_j$  for the  $j$ -th job is defined as 0 for  $j = 1$  and  $\max\{0, \max_{k \in [j-1]} \{\sum_{i=1}^k \max_{n \in [N]} S_{n,j-i} - \sum_{i=1}^k A_{j-i}\}\}$ , for  $j > 1$  (Rizk, Poloczek, and Ciucu 2015, 2016). The key observation here is that we can view the blocking FJ system with  $N$  servers as a hypothetical work-conserving system with a single server whose service times are distributed as  $S_i^* \stackrel{\mathcal{D}}{=} \max_{n \in [N]} S_{n,i}$ . This allows us to use results from Section 4.1 to analyse a blocking FJ system as a special case. We have the following steady-state representation of the waiting time  $W'$  for the blocking FJ system with  $N$  servers

$$W' \stackrel{\mathcal{D}}{=} \max_{k \in \mathbb{N}_0} Y_k \text{ with } Y_k := \sum_{i=1}^k Y_i^A, \quad (4.2.1)$$

where  $Y_i^A := \max_{n \in [N]} S_{n,i} - A_i$  for all  $i \in \mathbb{N}$  and set  $Y_0 := 0$ . Also define

$$\zeta_k(s) := k^{-1} \log \mathbb{E}[\exp(sY_k)], \text{ and } \zeta(s) := \lim_{k \rightarrow \infty} k^{-1} \log \mathbb{E}[\exp(sY_k)].$$

The upper bound on the tail probabilities of the steady-state waiting times then follows from Theorem 4.1.2 in a straightforward fashion. Therefore, we state the following corollary to Theorem 4.1.2 without a proof. The transformed kernel  $\tilde{L}$  is calculated using (4.1.5). For ease of implementation, the Algorithm 4.2 dedicated to the blocking case is also provided.

**Corollary 4.2.1** (Blocking systems). *Consider an FJ system with  $N$  parallel blocking servers, as described in Section 4.2. Then, we have*

1. *For all  $s \in \mathcal{D}\zeta$ ,  $\exp(\zeta(s))$  is the simple maximal eigenvalue of  $\tilde{L}$ , and the corresponding right eigenfunction  $\{r(c, s); c \in \mathbb{E}\}$  satisfying*

$$\exp(\zeta(s))r(c, s) = \int_{\mathbb{R}} \tilde{L}(c, d\tau; s)r(\tau, s),$$

*is positive and bounded above.*

2. *The tail probabilities of the steady-state waiting times defined in (4.2.1) are bounded above by*

$$\mathbb{P}(W' \geq w) \leq \phi(\theta) \exp(-\theta w), \quad (4.2.2)$$

*where  $\theta := \sup\{s > 0 \mid \zeta(s) \leq 0\}$  and  $\phi(s) := \text{ess sup}\{\mathbb{1}(Y_1 > 0)/r(C_1, s)\}$  after having normalised  $r(\cdot, \theta)$  so that  $\mathbb{E}[r(C_0, \theta)] = 1$ .*

**Algorithm 4.2** Pseudocode for blocking systems

---

**Require:** Transition kernel  $T$ , and the MGFs  $E_\tau \left( \exp(s \max_{n \in [N]} S_{n,i}) \right), E_\tau \left( \exp(-s A_1) \right)$

- 1: **if** A1 and A2 and A3 and A4 **then**
- 2:   Transform  $T$  to get  $\tilde{L}(c, d\tau; s)$  (see (4.1.5))
- 3:    $\exp(\zeta(s)) \leftarrow$  maximal eigenvalue of  $\tilde{L}(c, d\tau; s)$
- 4:    $\theta \leftarrow \sup\{s > 0 \mid \zeta(s) \leq 0\}$
- 5:   Normalise  $r(\cdot, \theta)$  so that  $E[r(C_0, \theta)] = 1$
- 6:    $\phi(\theta) \leftarrow \sup_{c \in \mathbb{E}} \{1/r(c, \theta)\}$
- 7: **end if**
- 8: **return** The decay rate  $\theta$  and the prefactor  $\phi$

---

In the following sections, we apply our results to several application scenarios. They are intended to serve as illustrative examples. For the sake of simplicity, assume that the state space  $\mathbb{E}$  of the chain  $\{C_k\}_{k \in \mathbb{N}_0}$  is finite. Then, the transition kernel  $T$  of  $\{C_k\}_{k \in \mathbb{N}_0}$  is just a transition matrix. Let us write  $T = ((t_{ij}))$ . We do allow the servers to follow different probability distributions satisfying stability conditions A3 and A4. For purposes of illustration, we consider exponentially distributed service and inter-arrival times in the following examples.

## 4.3 FORK-JOIN SYSTEM WITH NON-RENEWAL INPUT

In this section, we describe an FJ system with Markov-modulated inputs. This is principally motivated by recent empirical evidences that reveal burstiness in Internet traffic and also in inputs to MapReduce clusters (Y. Chen, Alspaugh, and Katz 2012; Heffes and Lucantoni 1986; Kandula et al. 2009; Yoshihara, Kasahara, and Takahashi 2001). In general, in order to model this burstiness, we can assume the inter-arrival times to be modulated by some Markov chain  $\{C_k\}_{k \in \mathbb{N}_0}$ .

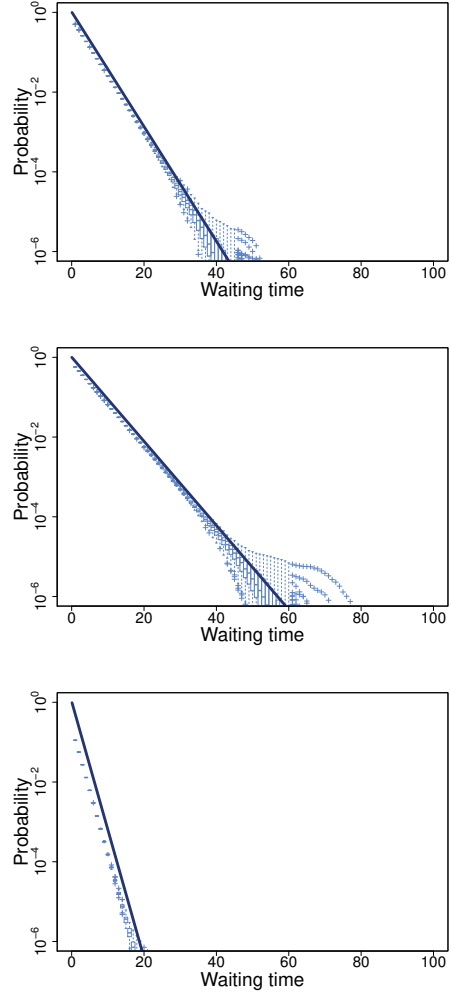
**Example 4.3.1** (Markov-modulated inter-arrival times). Suppose the modulating Markov chain takes four distinct values (corresponding to different phases of arrival traffic). In state  $j$  of the chain, suppose the inter-arrival times are exponentially distributed with parameter  $\lambda_j$ . Also assume, the service times at the  $n$ -th server are exponentially distributed with parameter  $\mu_n$ . Then, the transformation in (4.1.15) is given by

$$\begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} & t_{1,4} \\ t_{2,1} & t_{2,2} & t_{2,3} & t_{2,4} \\ t_{3,1} & t_{3,2} & t_{3,3} & t_{3,4} \\ t_{4,1} & t_{4,2} & t_{4,3} & t_{4,4} \end{pmatrix} \mapsto \begin{pmatrix} t_{1,1} \frac{\lambda_1}{\lambda_1+s} & t_{1,2} \frac{\lambda_2}{\lambda_2+s} & t_{1,3} \frac{\lambda_3}{\lambda_3+s} & t_{1,4} \frac{\lambda_4}{\lambda_4+s} \\ t_{2,1} \frac{\lambda_1}{\lambda_1+s} & t_{2,2} \frac{\lambda_2}{\lambda_2+s} & t_{2,3} \frac{\lambda_3}{\lambda_3+s} & t_{2,4} \frac{\lambda_4}{\lambda_4+s} \\ t_{3,1} \frac{\lambda_1}{\lambda_1+s} & t_{3,2} \frac{\lambda_2}{\lambda_2+s} & t_{3,3} \frac{\lambda_3}{\lambda_3+s} & t_{3,4} \frac{\lambda_4}{\lambda_4+s} \\ t_{4,1} \frac{\lambda_1}{\lambda_1+s} & t_{4,2} \frac{\lambda_2}{\lambda_2+s} & t_{4,3} \frac{\lambda_3}{\lambda_3+s} & t_{4,4} \frac{\lambda_4}{\lambda_4+s} \end{pmatrix}.$$

Having done the above transformation, the decay rates are found as

$$\begin{aligned} \theta_n &= \sup\{s > 0 \mid \frac{\mu_n}{\mu_n - s} \chi_A(s) \leq 1\}, \\ \theta &= \sup\{s > 0 \mid \beta(\mu; s) \chi_A(s) \leq 1\}, \end{aligned} \tag{4.3.1}$$

**Figure 4.3:** Numerical verification of the bounds (shown in darker shade) for blocking systems. **(Top)** An FJ system with Markov-modulated arrivals. The modulating Markov chain takes values in the set  $\mathbb{E} = \{1, 2, 3\}$ . The exponential inter-arrival times have parameters 0.25, 0.4, and 0.50. **(Middle)** An FJ system with Markov-modulated service times. The modulating Markov chain takes values in the set  $\mathbb{E} = \{1, 2, 3, \dots, 32\}$ . The exponential inter-arrival times have parameter 0.35. **(Bottom)** An FJ system with Markov-modulated arrival and service times. The modulating Markov chain takes values in the set  $\mathbb{E} = \{1, 2, 3, \dots, 64\}$ . In all the cases, there are five heterogeneous, blocking servers whose service rates are drawn randomly, satisfying the stability conditions in A3 and A4. The transition probabilities and the initial distribution of  $C_k$  are chosen randomly. Observe that the analytic bounds obtained in Corollary 4.2.1 are in close agreement with the sample estimates of the tail probabilities of the waiting times.



where  $\chi_A$  is the largest eigenvalue of the transformed matrix and

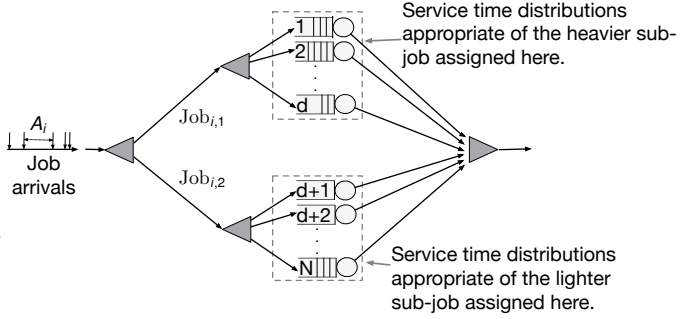
$$\beta(\mu; s) := \sum_{S \in \{A \subset [N] \mid A \neq \emptyset\}} (-1)^{|S|+1} \frac{(\sum_{i \in S} \mu_i)}{(\sum_{i \in S} \mu_i) - s}, \quad (4.3.2)$$

is the MGF of the maximum of  $N$  independent exponential random variables (see Appendix B.2). After normalisation of the right eigenvector, one obtains the bounds using (4.1.13) and (4.2.2) for the work-conserving and the blocking system respectively. Please see Figure 4.2 (for work-conserving systems) and Figure 4.3 (for blocking systems) to compare our bounds against CCDFs obtained from Monte Carlo simulations.

#### 4.4 PARALLEL SYSTEMS WITH DEPENDENT SERVERS

In this section, we consider an FJ system as described in Section 4.1 with correlated servers. To be precise, we assume that the service times are modulated by a Markov

**Figure 4.4:** Single-node FJ system with a provisioning where the heavier part of each incoming job is apportioned in a round-robin fashion. Let  $\text{Job}_i = (\text{Job}_{i,1}, \text{Job}_{i,2})$ , where  $\text{Job}_{i,1}$  denotes the heavier sub-job. For instance, for the first job  $\text{Job}_1$ , the sub-job  $\text{Job}_{1,1}$  is allotted to servers  $1, 2, \dots, d$  and  $\text{Job}_{1,2}$ , to the rest. Then,  $\text{Job}_{2,1}$  is allotted to servers  $d+1, d+2, \dots, 2d$  and  $\text{Job}_{2,2}$  to the rest, and so on.



chain. The motivation behind this is the phase-type behaviour that service times show due to various exogenous effects. Before furnishing numerical examples, we mention some factors that might engender such a phase-type behaviour.

**UNEQUAL JOB SIZES** Phase-type behaviour may arise when the sizes of the incoming jobs are unequal enforcing a change of service time distribution across the servers. Intuitively, heavier jobs demand greater service times in total. This can be modelled by scaling up the service times or the parameters of their distributions whenever a heavier job arrives. For instance, in the context of MapReduce, the job sizes can be time varying. In the context of Multi-path TCP, the packet sizes are usually of different sizes. The modulating chain captures the different job sizes enforcing different service time distributions. The state space of the chain  $\mathbb{E}$  can be chosen depending on the particular application under consideration.

**PROVISIONING IN MAPREDUCE** The “irregular” service times may also arise due to provisioning, even when the job sizes are constant. Suppose that the incoming jobs are split unequally among the available servers. The rule that decides job division into tasks is termed *provisioning*. Such provisioning can be employed in MapReduce systems to influence waiting times. Consider the following example: Each job consists of two sub-jobs one of which is more demanding than the other. That is,  $\text{Job}_i = (\text{Job}_{i,1}, \text{Job}_{i,2})$ , where  $\text{Job}_{i,1}$  can be assumed to be heavier (more time-consuming) without loss of generality. Now, in order to apportion the burden of the heavier job, devise a variant of round-robin mechanism such that for the first job  $\text{Job}_1$ , the sub-job  $\text{Job}_{1,1}$  is allotted to servers  $1, 2, \dots, d$  and  $\text{Job}_{1,2}$ , to the rest  $N - d$  servers. Then,  $\text{Job}_{2,1}$  is allotted to servers  $d+1, d+2, \dots, 2d$  and  $\text{Job}_{2,2}$  to the rest, and so on. Mathematically this is equivalent to having a modulating Markov chain that starts at state 1 where it assigns service time distributions appropriate of the heavier job (e.g., scaled service times as explained before) to servers  $1, 2, \dots, d$  and the usual unscaled service time distribution, to the rest, and then jumps with probability one to state 2 where it assigns service time distribu-

tions appropriate of the heavier job to servers  $d + 1, d + 2, \dots, 2d$  and the usual, to the rest. See Figure 4.4 for a pictorial description of this provisioning.

**MODULATION IN MULTI-PATH TCP** Packet scheduling or load balancing mechanisms (Frömmgen et al. 2017) could also give rise to correlated service times. The load balancing algorithm typically decides on the amount of packets to send over each path with the objective of keeping congestion under control. Taking the liberty of mathematical abstraction, we can model such a scenario with a Markov chain (representing the decisions of the load-balancer) that modulates only the service times of the system.

**EFFICIENCY DIFFERENTIATION** Servers may themselves have their own high and low efficiency periods that may or may not depend on the state of the other servers, e.g., enforced by energy-saving routines (Sharma et al. 2013). The service rates may also be modulated by the user. For instance, given a fixed monetary budget, the user of a cloud computing service such as the Amazon AWS, may be forced to switch to a less expensive machine (with inferior service rates) when the price of the current machines increase, to meet the budget constraint (e.g., see Shastri, Rizk, and Irwin (2016)).

**Example 4.4.1** (Markov-modulated service times). Motivated by the above scenarios, we now demonstrate the bound computation in (4.1.13) and (4.2.2). In the following example, assume the arrival process is renewal and inter-arrival times are exponentially distributed with parameter  $\lambda$ .

Suppose there are two servers each of which has two efficiency phases, high and low. We model this by two Markov chains modulating the servers, each on state space  $\{0, 1\}$ . For the sake of simplicity, assume that server  $i$  is exponentially distributed with parameter  $\mu_i$  or  $\kappa_i$  according as its modulating Markov chain is state 0 or 1. The two Markov chains may not be independent. Mathematically this is equivalent to having one single modulating Markov chain on state space  $\{0, 1\} \times \{0, 1\}$ . We rename the states as  $(0, 0) \mapsto 1, (0, 1) \mapsto 2, (1, 0) \mapsto 3, (1, 1) \mapsto 4$ .

Let us first look at the work-conserving system. For the 1st server, following (4.1.15), we transform

$$\begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} & t_{1,4} \\ t_{2,1} & t_{2,2} & t_{2,3} & t_{2,4} \\ t_{3,1} & t_{3,2} & t_{3,3} & t_{3,4} \\ t_{4,1} & t_{4,2} & t_{4,3} & t_{4,4} \end{pmatrix} \mapsto \begin{pmatrix} t_{1,1} \frac{\mu_1}{\mu_1 - s} & t_{1,2} \frac{\mu_1}{\mu_1 - s} & t_{1,3} \frac{\kappa_1}{\kappa_1 - s} & t_{1,4} \frac{\kappa_1}{\kappa_1 - s} \\ t_{2,1} \frac{\mu_1}{\mu_1 - s} & t_{2,2} \frac{\mu_1}{\mu_1 - s} & t_{2,3} \frac{\kappa_1}{\kappa_1 - s} & t_{2,4} \frac{\kappa_1}{\kappa_1 - s} \\ t_{3,1} \frac{\mu_1}{\mu_1 - s} & t_{3,2} \frac{\mu_1}{\mu_1 - s} & t_{3,3} \frac{\kappa_1}{\kappa_1 - s} & t_{3,4} \frac{\kappa_1}{\kappa_1 - s} \\ t_{4,1} \frac{\mu_1}{\mu_1 - s} & t_{4,2} \frac{\mu_1}{\mu_1 - s} & t_{4,3} \frac{\kappa_1}{\kappa_1 - s} & t_{4,4} \frac{\kappa_1}{\kappa_1 - s} \end{pmatrix}.$$

Transformation for the 2nd server is analogous. Denote the largest eigenvalues of these two transformed matrices by  $\chi_S^{(1)}$  and  $\chi_S^{(2)}$  respectively. The transformation for the blocking system is as follows

$$\begin{pmatrix} t_{1,1} & t_{1,2} & t_{1,3} & t_{1,4} \\ t_{2,1} & t_{2,2} & t_{2,3} & t_{2,4} \\ t_{3,1} & t_{3,2} & t_{3,3} & t_{3,4} \\ t_{4,1} & t_{4,2} & t_{4,3} & t_{4,4} \end{pmatrix} \mapsto \begin{pmatrix} t_{1,1}\beta(\mu_1, \kappa_1; s) & t_{1,2}\beta(\mu_1, \kappa_2; s) & t_{1,3}\beta(\mu_2, \kappa_1; s) & t_{1,4}\beta(\mu_2, \kappa_2; s) \\ t_{2,1}\beta(\mu_1, \kappa_1; s) & t_{2,2}\beta(\mu_1, \kappa_2; s) & t_{2,3}\beta(\mu_2, \kappa_1; s) & t_{2,4}\beta(\mu_2, \kappa_2; s) \\ t_{3,1}\beta(\mu_1, \kappa_1; s) & t_{3,2}\beta(\mu_1, \kappa_2; s) & t_{3,3}\beta(\mu_2, \kappa_1; s) & t_{3,4}\beta(\mu_2, \kappa_2; s) \\ t_{4,1}\beta(\mu_1, \kappa_1; s) & t_{4,2}\beta(\mu_1, \kappa_2; s) & t_{4,3}\beta(\mu_2, \kappa_1; s) & t_{4,4}\beta(\mu_2, \kappa_2; s) \end{pmatrix}.$$



Call its largest eigenvalue  $\chi_s$ . The function  $\beta$  is as defined in (4.3.2). Having done the above transformation, the decay rates are found as

$$\begin{aligned}\theta_n &= \sup\{s > 0 \mid \frac{\lambda}{\lambda+s} \chi_s^{(n)}(s) \leq 1\}, \\ \theta &= \sup\{s > 0 \mid \frac{\lambda}{\lambda+s} \chi_s(s) \leq 1\}.\end{aligned}\tag{4.4.1}$$

After normalisation of the right eigenvector, one finds the bounds on the tail probabilities of the steady-state waiting times using formulas in (4.1.13) and (4.2.2) for the work-conserving and the blocking system respectively. To see the quality of our bounds on a bigger state space, we simulated an FJ system with five heterogeneous servers being modulated by a chain having 32 states. See Figure 4.2 (for work-conserving systems) and Figure 4.3 (for blocking systems) to compare our bounds against empirical CCDFs.

#### 4.5 MARKOV MODULATED ARRIVALS AND SERVICE

In this section, we describe a system where service and inter-arrival times are dependent. This is essentially a generalization of Section 4.3 and Section 4.4. All the motivating examples listed in Section 4.3 and Section 4.4 can be extended to this case to account for generalised application scenarios. While this allows us to endow service times of each server, and the arrival process, separate modulating Markov chains (which can be modelled by one single chain on the Cartesian product space as shown before), we can use this formalism to devise more advanced provisioning by taking into account the current job arrival rate (*i.e.*, set efficiency of servers to “high” during busy period and to “low” otherwise etc.). This paves way for what we call “reactive provisioning.”

##### 4.5.1 Reactive provisioning

We propose to take into account information on the current FJ system environment, *e.g.*, estimates of the arrival intensities, and then modulate, *i.e.*, set service rates accordingly. Such a provisioning is reactive in nature and hence the nomenclature. The changing environment is essentially captured through the modulating Markov chain for the arrivals in this case.

**Example 4.5.1** (Markov-modulated inter-arrival and service times). Consider a Markov chain  $\{C_k\}_{k \in \mathbb{N}_0}$  capturing the changing environment in the sense that at state  $j$  of the Markov chain, the inter-arrival times are exponentially distributed with parameter  $\lambda_j$  and accordingly, the service times at the  $n$ -th server are distributed exponentially with parameter  $\mu_{n,j}$ . Define  $\mu^{(j)} := (\mu_{1,j}, \mu_{2,j}, \dots, \mu_{n,j})$ . Then, the required transformation for work-conserving systems is  $t_{ij} \rightarrow t_{ij} \left( \frac{\mu_{n,j}}{\mu_{n,j}-s} \right) \left( \frac{\lambda_j}{\lambda_j+s} \right)$ , for the  $n$ -th server, and likewise, the transformation for the blocking system is given by  $t_{ij} \rightarrow t_{ij} \beta(\mu^{(j)}; s) \left( \frac{\lambda_j}{\lambda_j+s} \right)$ . Let us denote the largest eigenvalue of the transformed matrix for the  $n$ -th server by  $\chi_{AS}^{(n)}$ , and

that of the transformed matrix for the blocking system by  $\chi_{AS}$ . Therefore, the decay rates are found as

$$\begin{aligned}\theta_n &= \sup\{s > 0 \mid \chi_{AS}^{(n)}(s) \leq 1\}, \\ \theta &= \sup\{s > 0 \mid \chi_{AS}(s) \leq 1\}.\end{aligned}\tag{4.5.1}$$

After normalisation of the right eigenvectors, we compute the bounds on the tail probabilities of the steady-state waiting times using formulas in (4.1.13) and (4.2.2). To see the quality of our bounds, we simulated the system with the modulating chain having 64 states. See Figure 4.2 for work-conserving systems and Figure 4.3 for blocking systems to compare our bounds against empirical CCDFs.

#### 4.6 FURTHER EXTENSIONS

In the following, we discuss how the results obtained in this chapter can be further extended to cover wider application scenarios.

##### 4.6.1 Design of Proactive Mechanisms

Markov-additive processes are capable of modelling not only reactive but also proactive systems. In Section 4.5, we modelled the changing environment with a Markov chain  $\{C_k\}_{k \in \mathbb{N}_0}$  and devised a reactive mechanism. For many applications, reactive mechanisms may be expensive, and it is profitable to be able to anticipate the changes in the environment and act accordingly (*e.g.*, set the service rates). Our Markov-additive process framework allows for such a proactive provisioning (see Figure 4.1). In this coupled model, the distribution of the increments  $X_{n,k+1}^A$ , for each  $n \in [N]$ , will also depend on  $C_k$ . Such a provisioning is promising as it allows for a notion of agility and adaptation in parallel server systems. The preparedness aimed for in proactive provisions could potentially reduce cost and yield a smoother transition.

##### 4.6.2 Replication with purging

Redundancy techniques have become increasingly popular over the last few years as a tool to decrease latency. For instance, in Vulimiri et al. (2013), the authors, based on empirical study, argue that redundancy can be effective in reducing latency in a large class of applications. The authors in Joshi, Soljanin, and Wornell (2017) model a cloud computing set-up as an FJ system with identical servers and study various redundancy techniques thoroughly. Such techniques typically create redundant tasks for each job with the hope of achieving smaller response times because creation of redundant jobs mitigates the synchronisation constraint at the output (see Figure 2.1) either entirely (in case of full replication) or partially (in case of partial replication, *e.g.*,  $(n, k)$  Fork-Join in Joshi, Soljanin, and Wornell (2017)). In Poloczek and Ciucu (2016), the authors discuss replication strategies and compute probability bounds on response times. Based on the bounds, they also devise a replication strategy that improves the stability region of classical FJ systems. In this section, we show that our Markov-additive framework for

a general FJ system developed in Section 4.1 can be immediately applied to study a purging replication strategy in an FJ system.

A purging replication strategy assigns (or replicates) each incoming job to each of the  $N$  available servers without splitting. Since there is no division of workload, there is no need to wait for all servers to finish executing their tasks. As such, a job leaves the system as soon as any of its  $N$  tasks, which are identical copies of the job itself, is executed. Therefore, there is no inherent synchronisation constraint at the output. Purging enforces that as soon as the first server executes its task, all other servers immediately discontinue their tasks at that time and take up the task of the next job, if available. The servers are therefore work-conserving. Although it appears to be different from the model described in Section 4.1 because of the absence of the output synchronisation, it can be viewed as a special case of our model by means of a simple analogy. As done in case of an FJ system with blocking servers in Section 4.2, an FJ system with  $N$  heterogeneous work-conserving servers governed by a purging replication strategy can be viewed as a hypothetical work-conserving system with a single server whose service times are now distributed as  $\tilde{S}_i \stackrel{\mathcal{D}}{=} \min_{n \in [N]} S_{n,i}$ . Therefore, the steady-state waiting time has the following representation

$$\tilde{W} \stackrel{\mathcal{D}}{=} \max_{k \in \mathbb{N}_0} Z_k \text{ with } Z_k := \sum_{i=1}^k Z_i^A, \quad (4.6.1)$$

where  $Z_i^A := \min_{n \in [N]} S_{n,i} - A_i$  for all  $i \in \mathbb{N}$  and set  $Z_0 := 0$ . Also define

$$\rho_k(s) := k^{-1} \log \mathbb{E}[\exp(sZ_k)], \text{ and } \rho(s) := \lim_{k \rightarrow \infty} k^{-1} \log \mathbb{E}[\exp(sZ_k)].$$

The upper bound on the tail probabilities of the steady-state waiting times can then be derived directly from Theorem 4.1.2. Therefore, we have the following corollary to Theorem 4.1.2. The transformed kernel  $\tilde{L}$  is calculated using (4.1.5).

**Corollary 4.6.1** (Replication with purging). *Consider an FJ system with  $N$  parallel work-conserving servers governed by a purging replication strategy. Then, we have*

1. *For all  $s \in \mathcal{D}\rho$ ,  $\exp(\rho(s))$  is the simple maximal eigenvalue of  $\tilde{L}$  and the corresponding right eigenfunction  $\{\tilde{r}(c, s); c \in \mathbb{E}\}$  satisfying*

$$\exp(\rho(s))\tilde{r}(c, s) = \int_{\mathbb{R}} \tilde{L}(c, d\tau; s)\tilde{r}(\tau, s),$$

*is positive and bounded above.*

2. *Tail probabilities of the steady-state waiting time defined in (4.6.1) are bounded above by*

$$\mathbb{P}(\tilde{W} \geq w) \leq \phi(\theta) \exp(-\theta w), \quad (4.6.2)$$

*where  $\theta := \sup\{s > 0 \mid \rho(s) \leq 0\}$  and  $\phi(s) := \text{ess sup}\{\mathbb{1}(Z_1 > 0)/\tilde{r}(C_1, s)\}$  after having normalised  $r(\cdot, \theta)$  so that  $\mathbb{E}[\tilde{r}(C_0, \theta)] = 1$ .*

**Example 4.6.1.** Suppose the arrival process is renewal, *i.e.*,  $\mathbb{E} = \{1\}$ . Then, instead of solving an eigenvalue problem, we solve a nonlinear equation involving the MGF and the Laplace transform of the service times and the inter-arrival times. Suppose the service times of the  $n$ -th server are exponentially distributed with rate  $\mu_n$ , for  $n \in [N]$ . Also assume the inter-arrival times are exponentially distributed with parameter  $\lambda$ . Since the minimum of a finite collection of exponential random variables is itself exponentially distributed, the decay rate  $\theta$  in Corollary 4.6.1 is found by solving the following equation

$$\left( \frac{\sum_{n \in [N]} \mu_n}{\sum_{n \in [N]} \mu_n - \theta} \right) \left( \frac{\lambda}{\lambda - \theta} \right) = 1,$$

which yields a closed-form solution  $\theta = \sum_{n \in [N]} \mu_n - \lambda$ . The upper bound on the tail probabilities is then found by plugging in this value of  $\theta$  in (4.6.2).

Interestingly, we would have obtained the same decay rate if we had one single server with combined capacity, *i.e.*, whose service times were exponentially distributed with rate  $\sum_{n \in [N]} \mu_n$ . Therefore, as far as the asymptotic decay rate of the tail probabilities of the steady-state waiting times is concerned, an FJ system with  $N$  work-conserving servers governed by a purging replication strategy has the same performance as a queueing system with one single server whose service rate is equal to the total of the individual service rates. It is remarkable that even a simple bound such as the one obtained in this example can present such insights into the performance of nontrivial FJ systems with replication strategies.

#### 4.6.3 Renewal Processes as a special case

Several previously known results on Fork-Join systems where a renewal arrival process was assumed (*e.g.*, the renewal cases in KhudaBukhsh, Rizk, et al. (2017) and Rizk, Poloczec, and Ciucu (2015), and also the FJ system in Chapter 3) can be retrieved by setting  $\mathbb{E} = \{1\}$ . In this case, following Algorithm 4.1 and Algorithm 4.2, the bounds turn out to be

$$P(W \geq w) \leq \sum_{n \in [N]} \exp(-\theta_n w), \text{ and } P(W' \geq w) \leq \exp(-\theta w), \quad (4.6.3)$$

where

$$\begin{aligned} \theta_n &= \sup\{s > 0 \mid E[\exp(sS_{n,1})]E[\exp(-sA_1)] \leq 1\}, \\ \theta &= \sup\{s > 0 \mid E[\exp(s \max_{n \in [N]} S_{n,1})]E[\exp(-sA_1)] \leq 1\}. \end{aligned}$$

This further enhances the applicability of our results.

In the next chapter, we shall apply the results obtained for general FJ systems in the previous and the current chapter to the collaborative uploading problem discussed in Section 1.1. In particular, we shall use the bounds to devise uploading strategies for this scenario.

## COLLABORATIVE UPLOADING

---

In this chapter, we consider the collaborative uploading problem described in Section 1.2. Our goal is to find optimal collaborative uploading strategies. We differentiate between the intermittent (devices such as sensors sending data on a coarse time scale) and the continuous collaborative uploading (devices continuously streaming video footage, *e.g.*, using Facebook Live (*Facebook Live* 2018), Periscope (Twitter, Inc 2018) ) cases and study them separately.

We analyse replication and allocation strategies for the collaborative uploading scenario. For the continuous stream uploading case, we use an FJ queueing system formulation that captures the ability to split data into chunks that are transmitted over multiple paths, and finally merged when all chunks are received. The results developed in Chapters 3 and 4 are utilised to design optimal strategies for the stream uploading case. We provide closed-form expressions for the mean upload latency in the intermittent uploading case, allowing a comparison between a replication and an allocation (splitting) strategy. We find optimal strategies for given path latencies. In doing so, we also show numerical results suggesting near-optimality of the proportional allocation.

### 5.1 MODELLING APPROACH

Here, we present an overview of our approach, which consists of (i) defining an appropriate performance metric, and (ii) framing an appropriate optimisation problem thereafter.

**THE INTERMITTENT CASE** We characterise the intermittent case as one where the time intervals between two successive uploads are so large that there is *no* self-induced queueing. Then, aspects such as cross-traffic can be described by means of the statistical properties of the path latencies alone. A primary device uploading data intermittently aims to minimise the upload latency, *i.e.*, the time until the data reaches the cloud. Given multiple paths, the primary device may split the data into chunks that are transmitted or replicated over the available paths. The upload latency being a stochastic quantity, it is natural to consider its mean as a performance metric and optimise it over all possible splitting/replication configurations. In Section 5.2, we express the upload latency as an order statistic of the individual upload times over the different paths, making the theory of order statistics a useful tool in our analysis.

**THE STREAM UPLOADING CASE** In the case of continuous upload of a data stream, *e.g.*, a primary device uploading a live video to the cloud, there is a notion of waiting before each data chunk can be uploaded and hence, that of queueing. We call the event of new data generation and passing by the application to the lower layers on the primary device, an arrival of a new data batch. Each data batch is split into chunks of various sizes that are transported over several paths. Paths are characterised by a random service

time required to transport the assigned chunks. Finally, the data batch reaches the cloud when all of its chunks are received. Therefore, such systems are naturally modelled as FJ queueing systems.

## 5.2 INTERMITTENT COLLABORATIVE UPLOADING

In the following we consider the intermittent uploading case of data of size  $K$  over  $N$  possibly heterogeneous paths (*e.g.*, sensor or monitoring devices uploading data on a coarse time scale). Assume that the data can be divided into  $N$  smaller chunks consisting of packets. Then, every  $\mathbf{k} = (k_1, k_2, \dots, k_N) \in \Lambda(N, K)$  is a valid allocation vector, where  $k_i$  denotes the number of packets to be sent via path  $i$  and  $\Lambda(N, K)$  denotes the set of all non-negative integer solutions to the Diophantine equation  $\sum_{i=1}^N k_i = K$ , for  $N, K \in \mathbb{N}$ . We denote the random amount of time taken to transport the  $j$ -th packet out of the  $k_i$  packets allocated to path  $i$  by  $D_{i,j}$ . Here,  $D_{i,j}$  may capture different phenomena that impact the transmission time over a path, such as resource allocation, transmission collisions, and retransmissions. Assume that for each  $i \in [N]$ , the random variables  $D_{i,j}$ 's are mutually independently distributed<sup>1</sup>. Recall that the data consisting of  $K$  packets can be reconstructed only after *all* the packets have arrived. Therefore, the upload latency can be expressed as  $D := \max(D_1^{(k_1)}, D_2^{(k_2)}, \dots, D_N^{(k_N)})$  where  $D_i^{(k_i)} := \sum_{j=1}^{k_i} D_{i,j}$  for  $k_i > 0$  denotes the amount of time taken by path  $i$  to transport  $k_i$  packets, and by convention,  $D_i^{(0)} := 0 \forall i \in [N]$ . The random variable  $D$  measures the total amount of time taken to transport *all* the packets over  $N$  different paths. In this work, we consider

$$\psi(\mathbf{k}) := \mathbb{E}[D] = \mathbb{E}[\max(D_1^{(k_1)}, D_2^{(k_2)}, \dots, D_N^{(k_N)})],$$

the expected upload time given an allocation  $\mathbf{k}$ , as our performance metric. The density function of  $D_i^{(k_i)}$  is given by the  $k_i$ -fold self-convolution of the density function of  $D_{i,j}$  due to independence. Let us denote the Cumulative Distribution Function (CDF) of  $D_i^{(k_i)}$  by  $F_i^{(k_i)}$ . Stacking into a column vector  $\mathbf{F}^{(\mathbf{k})} := (F_1^{(k_1)}, F_2^{(k_2)}, \dots, F_N^{(k_N)})^\top$ , we express the expected values of the order statistics of  $D_1^{(k_1)}, D_2^{(k_2)}, \dots, D_N^{(k_N)}$  as an operator  $\mu$  on  $\mathbf{F}^{(\mathbf{k})}$  (see Remark C.1.1 in Appendix C.1.1). Since  $\psi(\mathbf{k})$  is the first moment of the  $N$ -th order statistic, we get

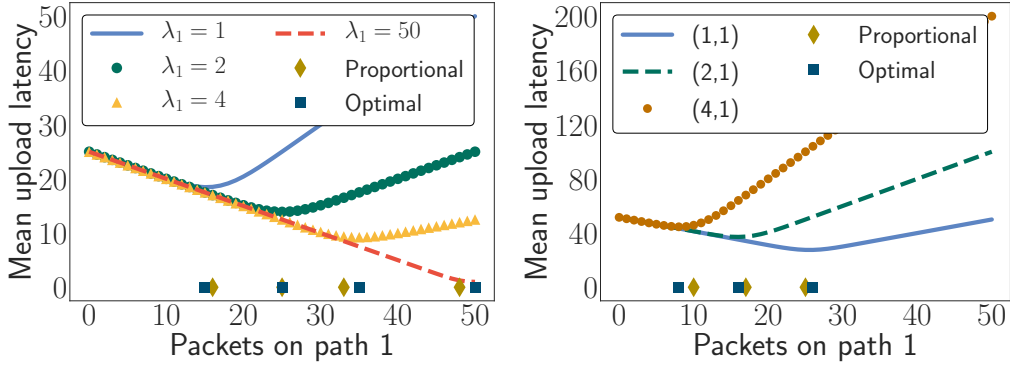
$$\psi(\mathbf{k}) = \mu_N \mathbf{F}^{(\mathbf{k})} = \sum_{j \in [N]} (-1)^{j+1} \mathbb{M}_j \mathbf{F}^{(\mathbf{k})}, \quad (5.2.1)$$

where  $\mu_N$  and  $\mathbb{M}_j$  are operators defined in Appendix C.1.1. The optimal allocation is found by minimising  $\psi$ , *i.e.*,

$$\mathbf{k}_{\text{opt}} := \underset{\mathbf{k} \in \Lambda(N, K)}{\operatorname{argmin}} \psi(\mathbf{k}). \quad (5.2.2)$$

Note that when the path characteristics are unknown, we can perform statistical inference. In the following, we show some illustrative examples with computable  $\mathbf{k}_{\text{opt}}$  before generalising this allocation scheme to include replication strategies.

<sup>1</sup> Mutual independence, although not necessary for the subsequent analysis, is assumed for the sake of simplicity. In order to account for possible dependencies observed in real-world applications, one needs to addi-



**Figure 5.1:** The canonical two-path case. We plot the mean upload latency as a function of the number of packets allocated to path 1 out of overall 50 packets. **(Left)** Both paths have exponentially distributed delays. The rate of the first path  $\lambda_1$  is increased from 1 to 50, while that of the second path is fixed at  $\lambda_2 = 2$ . Note the shift in the optimal allocation as  $\lambda_1$  increases. **(Right)** Path 1 has Weibull delay with (scale, shape) parameters given in the legend while path 2 has lognormal delay with parameters 0 and 0.25. Observe that the optimal allocation is indeed close to the proportional allocation.

### 5.2.1 The canonical two-path case

Let us consider the problem of finding the optimal allocation over two heterogeneous paths. Let  $\mathbf{k} \in \Lambda(2, K)$  denote our allocation. The corresponding upload latency is given by  $D := \max(D_1^{(k_1)}, D_2^{(k_2)})$  and its mean is given by

$$\psi(\mathbf{k}) = \mu_2 F^{(\mathbf{k})} = \mathbb{E}[D_1^{(k_1)}] + \mathbb{E}[D_2^{(k_2)}] - \int_0^\infty (1 - F_1^{(k_1)}(x))(1 - F_2^{(k_2)}(x)) dx. \quad (5.2.3)$$

Suppose the packet latencies  $D_{1,j}$  and  $D_{2,j}$  are exponentially distributed with rates  $\lambda_1$  and  $\lambda_2$ . Then, setting  $p = \frac{\lambda_1}{\lambda_1 + \lambda_2}$ ,  $q = 1 - p$ , and  $r = \frac{1}{\lambda_1 + \lambda_2}$ , the expected upload time is

$$\psi(\mathbf{k}) = \frac{k_1}{\lambda_1} + \frac{k_2}{\lambda_2} - r \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{k_2-1} \binom{n_1 + n_2}{n_1} p^{n_1} q^{n_2}.$$

Algebraic manipulation yields

$$\psi(k_1, k_2) \gtrless \psi(k_1 + 1, k_2 - 1) \iff \frac{I_p(k_1, k_2)}{I_{1-p}(k_2 - 1, k_1 + 1)} \gtrless \frac{\lambda_2}{\lambda_1},$$

where  $I_x(a, b)$  is the regularised  $\beta$ -function. This allows finding the optimal allocation  $\mathbf{k}_{\text{opt}}$  (see Appendix C.2). When  $K$  is large, the optimal strategy can be found by numerically solving the following nonlinear equation

$$\frac{I_p(x, K - x)}{I_{1-p}(K - x - 1, x + 1)} - \left(\frac{1}{p} - 1\right) = 0.$$

tionally specify a correlation structure for these variables. This step is application-specific and is not easy in general. We do not attempt that here.

In this case, the optimal allocation on path 1 is one of the two nearest integers producing a lower mean upload latency.

In Figure 5.1, we consider the canonical two-path scenario for different choices of path-specific delay distributions and show the mean upload latency as a function of the number of packets on path 1. For distributions not admitting a closed-form expression for the mean upload latency, *e.g.*, Weibull, lognormal, we performed numerical integration.

**Remark 5.2.1** (Near-optimality of proportional allocation: a comparison with Wen and Sun (2007) and G. Zhang et al. (2011)). The two-path scenario has been studied in Wen and Sun (2007) and G. Zhang et al. (2011) for the exponential delay model. The authors, however, do not compute a closed-form expression for the mean upload latency and only provide the following upper bound, based on a Chernoff technique

$$\psi(k_1, k_2) \leq \max \left\{ \frac{k_1}{\lambda_1}, \frac{k_2}{\lambda_2} \right\} + \sqrt{2\pi} \left( \sqrt{\frac{k_1}{\lambda_1^2}} + \sqrt{\frac{k_2}{\lambda_2^2}} \right) \text{ (due to G. Zhang et al. (2011)).}$$

Based on the above bound, the authors characterise the optimal allocation as being either the proportional allocation, *i.e.*,  $(k_1, k_2) = (Kp, K - Kp)$  or the winner-takes-it-all allocation, *i.e.*,  $(k_1, k_2) = (K, 0)$ . In contrast, we provide exact closed-form expression for the mean upload delay and find the optimal allocation  $\mathbf{k}_{\text{opt}}$ . Interestingly, we observe near-optimality of the proportional allocation, *e.g.*, as shown in Figure 5.2 (left) for exponential path delays. In Figure 5.1, we see that similar conclusions hold for Weibull and lognormal delays as well.

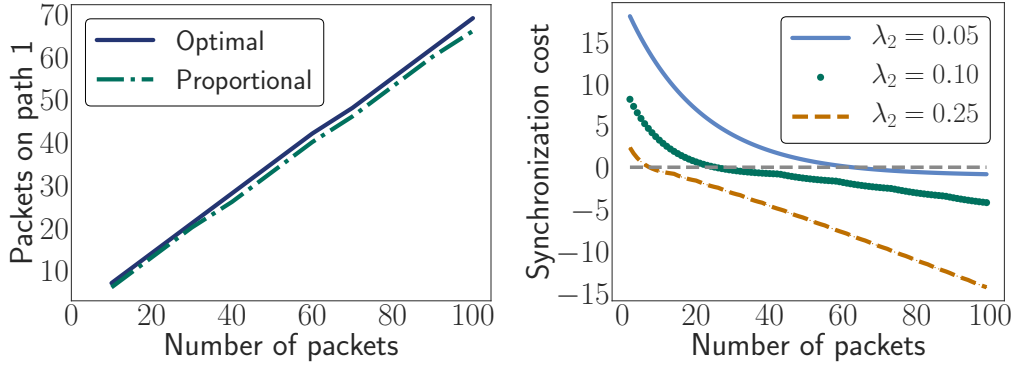
### 5.2.2 The $N$ -path case with exponential delays

We next consider the general case of  $N$  paths available for uploading  $K$  packets of data. Suppose the  $i$ -th path has exponential delay with rate  $\lambda_i$ . The mean upload latency of the allocation  $\mathbf{k} \in \Lambda(N, K)$  is given by

$$\psi(\mathbf{k}) = \sum_{S \in \{A \subseteq [N] : A \neq \emptyset\}} (-1)^{|S|+1} \sum_{0 \leq n_i \leq k_i - 1 : i \in S} \left( \prod_{i \in S} \frac{\lambda_i^{n_i}}{n_i!} \right) \times \frac{\Gamma(\sum_{i \in S} n_i + 1)}{(\sum_{i \in S} \lambda_i)^{\sum_{i \in S} n_i + 1}}.$$

The derivation is provided in Appendix C.2, where we also present additional numerical examples.





**Figure 5.2:** (Left) Near-optimality of proportional allocation. The number of packets allocated to path 1 versus the overall number of packets (data size) for two exponentially distributed path latencies with parameters  $\lambda_1 = 4$  and  $\lambda_2 = 2$ . (Right) The synchronisation cost as a function of data size. We consider two paths having exponential delays with rates  $\lambda_1 = 1$  and  $\lambda_2$  in the legend. Recall from (5.2.5) that positive (negative) synchronisation cost implies superiority of replication (allocation). For large data sizes, it is better to allocate than to replicate. However, if one of the paths is much slower compared to the other one, the synchronisation cost is high and consequently, replication may become profitable.

**Remark 5.2.2** (Upper bound on the mean upload latency). In certain situations the mean upload latency can not be obtained in closed-form. However, following Boucheron, Lugosi, and Massart (2013), we can obtain the following upper bound on the mean upload latency

$$\psi(k_1, k_2, \dots, k_N) \leq \inf_{y \in \cap_{i \in [N]} \mathcal{D}_{\kappa_i}} \frac{1}{y} \log \left( \sum_{i \in [N]} \kappa_i(y) \right), \quad (5.2.4)$$

where  $\kappa_i(y) := \mathbb{E}[\exp(yD_i^{(k_i)})]$ . As before, our strategy will be to minimise the right hand side of the above inequality.

The allocation strategies discussed so far inherently impose a synchronisation constraint at the destination. At a certain overhead, one way to circumvent this synchronisation constraint is replication, which we consider next.

### 5.2.3 Replication strategies

A basic replication strategy is to send the entire data over all available paths and take the first chunk that arrives at the destination. Replication strategies are known to reduce latency in some regimes (Vulimiri et al. 2013). However, an apparent drawback is their overuse of resources, *e.g.*, higher energy consumption. Roughly put, replication replaces the max operation (requiring the last chunk to arrive to complete the data at the receiver) with a min operation (taking the first to arrive at the receiver). However, the min op-

eration is taken over elements that stochastically dominate the elements over which the max operation is taken. This poses an interesting trade-off: *when should we replicate, and not allocate?*

In the basic replication case, the upload latency is  $D := \min(D_1^{(K)}, D_2^{(K)}, \dots, D_N^{(K)})$  where  $D_i^{(K)} := \sum_{j=1}^K D_{i,j}$ . Our objective remains minimising the mean upload latency

$$\phi(N, K) := \mathbb{E}[D] = \mathbb{E}[\min(D_1^{(K)}, D_2^{(K)}, \dots, D_N^{(K)})] = \mu_1 F^{(Kv)},$$

where  $v$  is an  $N$ -dimensional vector of all ones and  $F^{(Kv)} = (F_1^{(K)}, F_2^{(K)}, \dots, F_N^{(K)})$ . We favour the replication strategy if  $\phi(N, k)$  is smaller than the mean upload latency of *any allocation*  $k \in \Lambda(N, K)$ , i.e., if  $\phi(N, K) \leq \min_{k \in \Lambda(N, K)} \psi(k)$ . In relation to the question of replication versus allocation, we introduce next the notion of synchronisation cost.

**SYNCHRONISATION COST** Suppose all available paths are used for transmission and let  $\Lambda^*(N, K) := \{k = (k_1, k_2, \dots, k_N) \in \Lambda(N, K) \mid k_i > 0 \ \forall \ i \in [N]\}$  denote the reduced set of valid allocations. Within  $\Lambda^*$ , an allocation can be worse than a replication essentially because of the synchronisation at the destination, i.e., because of some paths being much slower than others. In order to compare with a replication strategy, we define the synchronisation cost given  $N$  paths and data size  $K$  as

$$\chi(N, K) := \min_{k \in \Lambda^*(N, K)} \psi(k) - \phi(N, K) = \min_{k \in \Lambda^*(N, K)} \mu_N F^{(k)} - \mu_1 F^{(Kv)}. \quad (5.2.5)$$

If  $\chi$  is positive, replication yields smaller mean upload latency and hence, is preferred. If  $\chi$  is negative, we prefer allocation over replication. Intuitively, if the data size is large, we expect the cost of redundancy to be high and  $\chi$  to be negative.

Consider the canonical two-path example with exponential delays from Section 5.2.1. A straightforward computation of  $\mu_1 F^{(Kv)}$  yields the following closed-form expression of the synchronisation cost defined in (5.2.5),

$$\chi(2, K) = \min_{(k_1, k_2) \in \Lambda^*(2, K)} \psi(k_1, k_2) - r \sum_{n_1=0}^{K-1} \sum_{n_2=0}^{K-1} \binom{n_1 + n_2}{n_1} p^{n_1} q^{n_2}.$$

In Figure 5.2, we show the synchronisation cost as a function of the data size  $K$ . As the data size increases the cost of redundancy worsens the performance of replication. Consequently, an allocation strategy is preferred for large data. However, the *zero-crossing* data size seen in Figure 5.2, which marks the regimes where replication and allocation are more beneficial, shifts depending on path heterogeneity.

#### 5.2.4 Combined allocation and replication: an $(N, r)$ -strategy

Here, we present a variant of the replication strategy, called the  $(N, r)$ -strategy. An  $(N, r)$ -strategy splits data of size  $K$  into  $N$  smaller chunks so that the data batch can be reconstructed from any  $r$  out of the  $N$  chunks. One of the ways to achieve such a splitting is to use Erasure codes, e.g., Maximum Distance Separable (MDS) codes (Joshi,

Soljanin, and Wornell 2017). Note that an  $(N, N)$ -strategy corresponds to allocation and an  $(N, 1)$ -strategy, to replication. In order to formulate an  $(N, r)$ -strategy, we define

$$Y(N, r, K) := \{k \in [K]^N \mid \sum_{i \in S} k_i \geq K \forall S \subseteq [N], |S| = r\}.$$

We call a  $k \in Y(N, r, K)$  an  $(N, r)$ -allocation for data of size  $K$ . The data is received as soon as the first  $r$  out of  $N$  chunks arrive at the destination. Let the order statistics corresponding to  $D_1^{(k_1)}, D_2^{(k_2)}, \dots, D_N^{(k_N)}$  be denoted by  $C_1 \leq C_2 \leq \dots \leq C_N$ . The mean upload latency for  $k \in Y(N, r, K)$  is given by

$$\eta_r(k) := E[C_r] = \mu_r F^{(k)}. \quad (5.2.6)$$

**Example 5.2.1** (Example of an  $(N, r)$ -strategy). Suppose we have three paths with exponential delays with parameters  $\lambda_1, \lambda_2$  and  $\lambda_3$ . Define, for  $i = 1, 2, 3$ ,  $p_{ij} = \frac{\lambda_i}{\lambda_i + \lambda_j}$ ,  $q_{ij} = 1 - p_{ij}$ ,  $r_{ij} = \frac{1}{\lambda_i + \lambda_j}$  and  $p_{123}^{(i)} = \frac{\lambda_i}{\lambda_1 + \lambda_2 + \lambda_3}$ ,  $r_{123} = \frac{1}{\lambda_1 + \lambda_2 + \lambda_3}$ . The mean upload latency corresponding to a  $(3, 1)$ -allocation (replication)  $k = (k_1, k_2, k_3) \in Y(3, 1, K)$  is given by

$$\mu_1 F^{(k)} = \eta_1(k) = r_{123} \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{k_2-1} \sum_{n_3=0}^{k_3-1} \frac{(n_1 + n_2 + n_3)!}{n_1! n_2! n_3!} \left(p_{123}^{(1)}\right)^{n_1} \left(p_{123}^{(2)}\right)^{n_2} \left(p_{123}^{(3)}\right)^{n_3}.$$

For a  $(3, 2)$ -allocation  $k \in Y(3, 2, K)$ , the mean upload latency,  $\mu_2 F^{(k)}$  is given by

$$\begin{aligned} \mu_2 F^{(k)} &= \mathbb{M}_2 F^{(k)} - 2\mathbb{M}_3 F^{(k)} \\ &= r_{12} \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{k_2-1} \frac{(n_1 + n_2)!}{n_1! n_2!} p_{12}^{n_1} q_{12}^{n_2} + r_{23} \sum_{n_2=0}^{k_2-1} \sum_{n_3=0}^{k_3-1} \frac{(n_2 + n_3)!}{n_2! n_3!} p_{23}^{n_2} q_{23}^{n_3} \\ &\quad + r_{31} \sum_{n_3=0}^{k_3-1} \sum_{n_1=0}^{k_1-1} \frac{(n_3 + n_1)!}{n_3! n_1!} p_{31}^{n_3} q_{31}^{n_1} - 2\eta_1(k). \end{aligned}$$

Note that a  $(3, 3)$ -allocation corresponds to simple allocation (see Section 5.2.2 and Khudabukhsh, Alt, et al. (2017)). The derivations are provided in Appendix C.2.

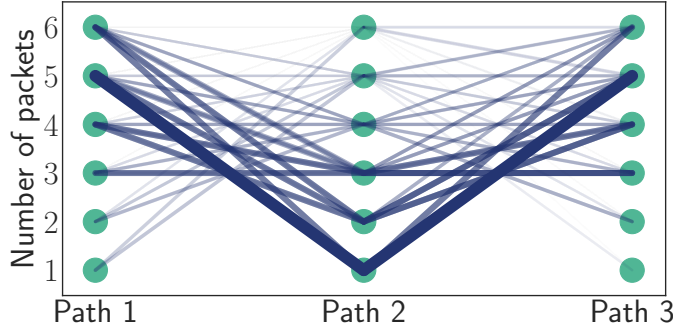
For a fixed  $r$ , the optimal  $(N, r)$ -allocation is given by  $k_{\text{opt}}^{(r)} := \arg\min_{k \in Y(N, r, K)} \eta_r(k)$ . We can, however, further improve the performance by optimising over  $r$ . In order to measure the performance of an allocation  $k$  compared to the optimal one, we define the regret of an  $(N, r)$ -allocation  $k$  as

$$\gamma(k) := \eta_r(k) - \min_{r \in [N]} \eta_r(k_{\text{opt}}^{(r)}). \quad (5.2.7)$$

In Figure 5.3, we consider three heterogeneous paths with exponential delays. We find the optimal allocation by minimising the regret. Interestingly, the optimal allocation is neither a  $(3, 1)$  replication, nor a  $(3, 3)$  allocation, but rather a  $(3, 2)$ -allocation.

### 5.3 STREAM UPLOADING

Now, we analyse collaborative uploading for continuous data streams using an FJ queueing model. An example scenario is the continuous upload of video data using multiple paths. We consider a rigid allocation strategy based on the probabilistic bounds on the steady-state waiting times derived in Chapters 3 and 4.



**Figure 5.3:** Optimal allocation by minimising regret: the lines specify different  $(3,2)$ -allocations. For example, the line joining 1, 5, and 6 corresponds to the allocation  $(1,5,6)$ . Valid  $(3,2)$ -allocations require the combined size of any 2 chunks to be at least the data size, here, 6. The darkness/thickness of the shades is inversely proportional to the regrets defined in (5.2.7). The allocation  $(5,1,5)$  (the thickest line), achieves zero regret and hence, is the optimal one. We assume exponential delays with rates 1, 5 and 10 in order of increasing path indices.

### 5.3.1 Rigid allocation based on steady-state bounds

We define the waiting time of an incoming data batch as the amount of time it waits until the last of its chunks starts getting uploaded. Consider the steady-state waiting time  $W$ . Following the work in Chapters 3 and 4, for a given allocation  $k \in \Lambda(N, K)$  and independent service times, we get

$$P(W \geq \sigma) \leq \exp(-\tilde{\theta}\sigma) \sum_{i \in [N]} \exp(-(\theta_i - \tilde{\theta})\sigma), \quad (5.3.1)$$

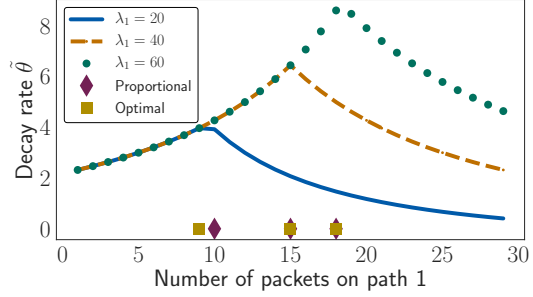
where  $\theta_i > 0$  is given by a condition involving the Laplace transforms of the inter-arrival times and the service times for  $k_i$  packets and  $\tilde{\theta} := \min_{i \in [N]} \theta_i$ . See Theorem 3.1.1. Here,  $\tilde{\theta}$  is the effective decay rate of the tail probability in the sense of an LDP, and assesses the quality of a given allocation (the higher the decay rate, the better).

Reducing the waiting times is equivalent to maximising the effective decay rate. Treating  $\tilde{\theta}$  as a function of the allocation, the optimal allocation is given by

$$k_{\text{opt}} := \underset{k \in \Lambda(N, K)}{\operatorname{argmin}} \tilde{\theta}(k). \quad (5.3.2)$$

**Example 5.3.1** (The canonical two-path scenario). Suppose we have two heterogeneous paths with exponential delays. Let the rates of the exponential distributions be  $\lambda_1$  and  $\lambda_2$ . Also, suppose the arrival process is renewal with exponentially distributed inter-arrival times. Let the rate of the inter-arrival distribution be  $\lambda_a$ .

**Figure 5.4:** Canonical two-path scenario for collaborative stream uploading. The effective decay rate  $\tilde{\theta}$  from (5.3.1) as a function of the number of packets sent via path 1 out of overall 30 packets. Both paths have exponential delays. We vary the rate of the first path  $\lambda_1$  as shown in the legend and keep the rate of the second path fixed at  $\lambda_2 = 40$ . The inter-arrival times are exponentially distributed with rate 0.5. Here too, we observe near-optimality of the proportional allocation.



Consider an allocation  $\mathbf{k} := (k_1, k_2) \in \Lambda(2, K)$ . Then,  $S_{1,1}^{(k_1)}$  and  $S_{2,1}^{(k_2)}$  are gamma distributed with shape and scale parameters  $(k_1, \lambda_1)$  and  $(k_2, \lambda_2)$  respectively. Then  $\theta_1$  and  $\theta_2$  are the solutions of the following two equations

$$\begin{aligned} \left(1 - \frac{\theta_1}{\lambda_1}\right)^{-k_1} \left(1 + \frac{\theta_1}{\lambda_a}\right) &= 1, \\ \left(1 - \frac{\theta_2}{\lambda_2}\right)^{-k_2} \left(1 + \frac{\theta_2}{\lambda_a}\right) &= 1. \end{aligned}$$

Solving the above two equations, we get the effective decay rate as

$$\tilde{\theta} = \min(\theta_1, \theta_2). \quad (5.3.3)$$

In Figure 5.4, we show how the effective decay rate depends on the data size  $K$ . Plotting the effective decay rate  $\tilde{\theta}$  as a function of the number of packets on path 1, we find the optimal allocation (yielding the largest decay rate). We also observe the near-optimality of the proportional allocation. In Appendix C.3, we also consider the two-path scenario with non-exponential path latencies. See Figure C.2.

**STREAM UPLOADING IN CHANGING ENVIRONMENTS** The approach presented above can be easily extended to account for changing environments. In order to account for the changing environments, we can model the FJ system representing the stream uploading scenario as a Markov-additive process as we did in Chapter 4. To be specific, we assume the path latencies are Markov-modulated by an exogenous Markov chain<sup>2</sup>. Please note that the definition of the waiting times remains the same as before. However, the bound on the tail probabilities of the steady-state waiting time changes. From (4.1.13) in Theo-

<sup>2</sup> The results developed in Chapter 4 allow us to modulate the inter-arrival as well as service times by an exogenous Markov chain. For our collaborate uploading problem, the inter-arrival times correspond to data generation times, which need to be modelled to be modulated by an exogenous Markov chain. Therefore, we only assume the path latencies are modulated by the Markov chain.

rem 4.1.2, we get the following upper bound on the tail probabilities of the steady-state waiting times  $W$ ,

$$P(W \geq w) \leq \sum_{n \in [N]} \phi_n(\theta_n) \exp(-\theta_n w), \quad (5.3.4)$$

where the quantities  $\theta_n$ ,  $\phi_n$  are as described in Theorem 4.1.2. Once we have obtained the above bound, we can carry out the optimisation as before to generate the optimal stream uploading strategy.

In this chapter, we optimised allocation and replication strategies for the collaborative uploading scenario described in Section 1.1. For the intermittent uploading case, we provided closed-form expressions for the mean upload latency. We posed the continuous stream uploading case as an FJ queueing model with varying burstiness of the data traffic to be uploaded, and of the paths' service. Optimal strategies are obtained by minimising the upper bounds on the tail probabilities of the steady-state waiting times derived in Chapters 3 and 4. Having obtained optimal collaborative uploading strategies for both the intermittent as well as the stream uploading cases, we shall next focus on two special class of queueing systems in the next two chapters of the thesis. In particular, we shall now relax the output synchronisation that is inherent to the FJ queueing systems. In the next chapter, we shall study parallel queueing systems with finite buffers and discuss preliminary ideas on optimal probabilistic scheduling. We shall provide a prefatory formulation of a parallel queueing system with exogenous modulation. The main theme in the Chapters 6 and 7 will be the use of random time change representation of Markov processes.

A significant proportion of classical queueing theory literature is concerned with the stability of queueing systems. In a stable queueing system, queue lengths do not explode to infinity, but are nevertheless allowed to be unbounded stochastic processes. One often looks for technical assumptions on the arrival process (relative to the service processes) that ensure stability of the queueing system. This is often achieved by establishing positive recurrence of the queue length process.

In real-life applications of queueing theory, the assumption of unbounded queues is often not tenable. In such a situation, we say the queueing system has a *finite* buffer, borrowing the term from communication networks. Let us use the term “buffer length” to denote the capacity of the buffer. For example, if the buffer length is some integer  $K$ , customers arriving when the present queue length is  $K$  are not added to the queue and are permanently lost (see Figure 6.1). Drawing analogy to communication networks again, we call the lost customers “dropped packets”.

The main objective of this preliminary study is to analyse transient queueing systems with finite buffers. We present a prefatory formulation of probabilistic scheduling in finite-buffer queueing systems under exogenous modulation. Finally, we present a scaling limit of finite-buffer queueing systems when the number of servers increases to infinity for a Cost-Based Queue-Aware (CBQA) randomised scheduling algorithm, called Join-Minimum-Cost (JMC) scheduling algorithm, which is a generalisation of the randomised job assignment scheme discussed in Mukhopadhyay, Karthik, and Mazumdar (2016).

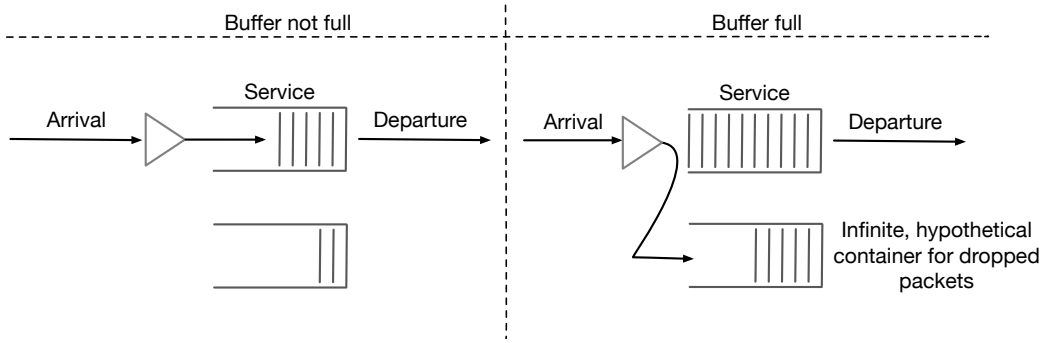
## 6.1 MODEL

We shall first study a single-server queueing system before generalising to multiple-server systems. We shall use random time change representation (Ethier and Kurtz 1986, Chapter 6) throughout for the key stochastic processes. Let  $\mathcal{T} := [0, T]$ , for some  $T > 0$ , be the fixed time interval of interest.

### 6.1.1 Single-server queueing systems

Let us consider a single-server queueing system with a finite buffer. Let  $K \in \mathbb{N}$  be the buffer length. Let the arrival process be a Poisson process with intensity  $\lambda$  and the service times be distributed as exponential random variables with rate  $\mu$ . The services times and inter-arrival times are assumed to be independent of each other and among themselves. Therefore, the queue length process  $X$  can be represented as

$$X(t) = X(0) + Y_1\left(\int_0^t \alpha(X(s))ds\right) - Y_{-1}\left(\int_0^t \beta(X(s))ds\right), \quad (6.1.1)$$



**Figure 6.1:** Description of the infinite hypothetical container. We consider a single-server queueing set-up with a finite buffer. Dropped packets are assumed to be accumulated in a virtual container with infinite capacity. Note that the virtual container is not a server. Therefore, there is no departure from the container.

where  $Y_1, Y_{-1}$  are independent unit-rate Poisson processes and the intensities  $\alpha, \beta : \mathbb{N}_0 \rightarrow \mathbb{R}_+$  are given by

$$\alpha(x) := \lambda \mathbb{1}(x < K) \quad (6.1.2)$$

$$\beta(x) := \mu \mathbb{1}(0 < x \leq K). \quad (6.1.3)$$

In order to model the loss process, we assume there is a hypothetical container where all the dropped packets are accumulated. The hypothetical container is assumed to have infinite capacity. Whenever the buffer is full, *i.e.*, the queue length is  $K$ , all incoming packets are lost. Lost packets are assumed to be accumulated in the hypothetical container.

The total loss process  $L$  keeps track of the cumulative count of dropped packets till a given instant of time. It is essentially the “queue length” at the hypothetical container. From a mathematical perspective, it behaves like a pure birth process, whose birth rate depends on an exogenous factor (the queue length). Therefore, it satisfies the following stochastic equation

$$L(t) = L(0) + Y_0\left(\int_0^t \gamma(X(s)) ds\right), \quad (6.1.4)$$

where  $Y_0$  is a unit Poisson process and the intensity  $\gamma : \mathbb{N}_0 \rightarrow \mathbb{R}_+$  is given by

$$\gamma(x) := \lambda \mathbb{1}(x = K). \quad (6.1.5)$$

Note that the process  $L$  itself is not Markovian, but the joint process  $(X, L)$  is Markovian on the state space  $\{0, 1, 2, \dots, K\} \times \mathbb{N}_0$ . The expected total loss can be found as a solution to the following integral equation

$$\mathbb{E}[L(t)] = \mathbb{E}[L(0)] + \int_0^t \lambda P(X(s) = K) ds \quad (6.1.6)$$

We can use (6.1.6) for performance evaluation purposes. In particular, we can treat the service rate  $\mu$  as our control variable and pose an optimal control problem for the queueing system.



Note that the probability  $P(X(s) = K)$  can be obtained by solving the Kolmogorov equations or the CMEs associated with  $(X, L)$ . Writing  $p_t(x, y) := P(X(t) = x, L(t) = y)$ , the CMEs are given by

$$\frac{d}{dt}p_t(x, y) = \begin{cases} \gamma(x)p_t(x, y-1) + \alpha(x-1)p_t(x-1, y) \\ \quad + \beta(x+1)p_t(x+1, y) - (\alpha(x) + \beta(x) + \gamma(x))p_t(x, y) & \text{if } x, y \geq 1, \\ \alpha(x-1)p_t(x-1, 0) + \beta(x+1)p_t(x+1, 0) \\ \quad - (\alpha(x) + \beta(x) + \gamma(x))p_t(x, 0) & \text{if } x \geq 1, y = 0, \\ \mu p_t(1, y) - \lambda p_t(0, y) & \text{if } x = 0, y \geq 0. \end{cases}$$

Marginalising out  $L$ , and writing  $q_t(x) := P(X(t) = x)$ , we only need to solve the following ODEs

$$\frac{d}{dt}q_t(x) = \begin{cases} \mu q_t(1) - \lambda q_t(0) & \text{if } x = 0, \\ \lambda q_t(x-1) + \mu q_t(x+1) - (\lambda + \mu)q_t(x) & \text{if } x = 1, 2, \dots, K-1, \\ \lambda q_t(x-1) - (\lambda + \mu)q_t(x) & \text{if } x = K. \end{cases}$$

**Remark 6.1.1.** In practical applications, the buffer lengths are often not prohibitively large. Therefore, solving the above  $(K+1)$  ODEs is usually feasible. In fact, the stationary probabilities can also be calculated analytically. Note that the present single-server queueing system with finite buffer is the same as the  $M/M/1/K+1$  model for which the above computations can also be carried out without the random time change representation (see Bolch et al. (2006, Chapter 6), for example). However, the random time change representation is often very convenient to work with analytically, especially for proving asymptotic results. We shall explore those possibilities later on.

### 6.1.2 $N$ -server queueing system

Suppose we have  $N$  servers available. Let  $K_i$  denote the buffer length of the  $i$ -th queue, for  $i \in [N]$ . Let  $X_i(t)$  denote the queue length at the  $i$ -th buffer, for  $i \in [N]$ . For this  $N$ -server queueing system, we additionally have a notion of scheduling, which we assume is probabilistic. Let  $\pi := (\pi_1, \pi_2, \dots, \pi_N)$  be a probability vector, i.e.,  $\pi_i \geq 0$  for all  $i \in [N]$  and  $\pi_1 + \pi_2 + \dots + \pi_N = 1$ . When *all* buffers are not full, an incoming packet is allocated to the  $i$ -th server with probability  $\pi_i$ . However, when a subset of the buffers are full, we modify the probabilities so that packets are dropped only when *all* buffers are full. The modification of server selection probability is done by removing from consideration the buffers that are full and apportioning unity into only the buffers that are not full. Therefore, if the  $i$ -th server is not full, we change its selection probability from  $\pi_i$  to  $\pi_i / \sum_{j: \mathbb{1}(X_j < K_j)} \pi_j$ . In particular, if the  $i$ -th server is the only one that is not full, an incoming packet is allocated to the  $i$ -th server with probability one. Let  $\mu_i$  denote the service rate of the  $i$ -th server.

As before, we denote the total loss process by  $L$ . Write  $X(t) = (X_1(t), X_2(t), \dots, X_N(t))$ . Then,  $(X, L)$  satisfies the following stochastic equations

$$X_i(t) = X_i(0) + Y_{1,i}(\int_0^t \alpha_i(X(s))ds) - Y_{-1,i}(\int_0^t \beta_i(X(s))ds), \quad (6.1.7)$$

$$L(t) = L(0) + Y_0(\int_0^t \gamma(X(s))ds), \quad (6.1.8)$$

where  $Y_{1,i}$ ,  $Y_{-1,i}$  and  $Y_0$  are independent unit Poisson processes and the intensities  $\alpha_i, \beta_i, \gamma : \mathbb{N}_0^N \rightarrow \mathbb{R}_+$  are defined as follows

$$\alpha_i(x) = \begin{cases} \lambda \pi_i / \sum_{j \in [N]} \pi_j \mathbb{1}(x_j < K_j) & \text{if } x_i < K_i, \\ 0 & \text{otherwise,} \end{cases} \quad (6.1.9)$$

$$\beta_i(x) = \mu_i \mathbb{1}(0 < x_i \leq K_i), \quad (6.1.10)$$

$$\gamma(x) = \lambda \mathbb{1}(x = (K_1, K_2, \dots, K_N)). \quad (6.1.11)$$

The expected total loss satisfies the following integral equation

$$\mathbb{E}[L(t)] = \mathbb{E}[L(0)] + \int_0^t \lambda \mathbb{P}(X(s) = (K_1, K_2, \dots, K_N))ds. \quad (6.1.12)$$

In order to compute  $\mathbb{E}[L(t)]$  using (6.1.12), we first compute the probability  $\mathbb{P}(X(s) = (K_1, K_2, \dots, K_N))$ , which we compute, as before, by solving the corresponding CMEs. In Appendix D.1, we show an example.

## 6.2 OPTIMAL SCHEDULING IN $N$ -SERVER QUEUES WITH FINITE BUFFERS

In this section, we discuss how the probabilistic scheduling in the  $N$ -server queueing system with finite buffers can be optimised. The main idea is to minimise the expected total loss, which serves as our performance metric in this work. Therefore, we look for a probability vector  $\pi$  that minimises the total expected loss. That is, the optimal probabilistic scheduling corresponds to

$$\pi_{\text{opt}} := \underset{\pi}{\operatorname{argmin}} \mathbb{E}[L(T)], \quad (6.2.1)$$

treating  $\mathbb{E}[L(T)]$  as a function of  $\pi$ . Standard gradient descent-type optimisation methods can be used to compute  $\pi_{\text{opt}}$  numerically.

**Remark 6.2.1** (Extension to other known scheduling algorithms). Deterministic schedules are naturally a special case of the probabilistic schedule. Moreover, note that the function  $\alpha_i$ 's can be appropriately modified to reflect other known scheduling algorithms. For instance, making  $\alpha_i$  one when the queue length at the  $i$ -th server is the shortest of all queue lengths, *i.e.*,  $X_i = \min\{X_1, X_2, \dots, X_N\}$  enables us to incorporate the Join-Shortest-Queue (JSQ) routine. Similarly, introducing a hierarchical rule that first looks for empty buffers and apportions unity among buffers that are empty will enable us to incorporate the Join-Idle-Queue (JIQ) routine. In fact, by choosing the  $\alpha_i$ 's appropriately, we can design various innovative scheduling algorithms, such as a mixed strategy by combining two or more different scheduling strategies.

### 6.3 QUEUEING SYSTEMS WITH EXOGENOUS MODULATION

In this section, we discuss finite-buffer queueing systems under the influence of exogenous factors, which we call the “environment”. Recent evidences suggest that the assumption of Markovianness is indeed untenable for several reasons. Arrival processes such as the input to a MapReduce system or datacentre traffic may exhibit considerable burstiness (Y. Chen, Alspaugh, and Katz 2012; Heffes and Lucantoni 1986; Kandula et al. 2009; Yoshihara, Kasahara, and Takahashi 2001). For the purpose of mathematical modelling, we assume the modulating environment is itself a Markov chain that modulates the arrival and the service processes.

Let  $C$  be a CTMC on a measurable space  $(\mathbb{E}, \mathcal{E})$ . For the sake of simplicity, let us assume the state space  $\mathbb{E}$  is finite with  $|\mathbb{E}| = M$ , for some  $M \in \mathbb{N}$ . We specify the intensities of jumps of  $C$  as follows

$$P(C(t + \delta t) - C(t) = \eta_i \mid \mathcal{F}(t)) = \kappa_i(C(t))\delta t + o(\delta t), \quad (6.3.1)$$

where  $\mathcal{F}(t)$  represents the history of the process (filtration generated by  $C$  over the time interval  $[0, t]$ ), the intensities  $\kappa_i$ ’s depend on the current state of  $C$ ,  $\eta_i$ ’s are the jump sizes, and  $\delta t > 0$ . There are  $M(M - 1)$  different types of jumps, to each of which we can assign a separate counting process. Therefore, the process  $C$  can be characterised as a solution to the stochastic equation

$$C(t) = C(0) + \sum_i \eta_i Y_{i,C} \left( \int_0^t \kappa_i(C(s)) ds \right), \quad (6.3.2)$$

where  $Y_{i,C}$ ’s are independent, unit Poisson processes. To be precise, the counting process  $Y_{i,C}$  keeps track of the number of jumps of type  $\eta_i$ .

#### 6.3.1 Modulation of the queueing system

We assume the intensity of the arrival process and the service rate are modulated by the chain  $C$ . Suppose the arrival intensity is  $\lambda_c$  when the chain  $C$  is in state  $c$ , for some  $c \in \mathbb{E}$ . Consider the  $N$ -server queueing system with buffer lengths  $K_1, K_2, \dots, K_N$ . Let  $\pi = (\pi_1, \pi_2, \dots, \pi_N)$  denote the vector of server selection probabilities. We assume the service rate of the  $i$ -th server is  $\mu_{i,c}$  when the chain  $C$  is in state  $c$ . We also assume that the arrival and the service processes are independent conditional on the chain  $C$ . Therefore, the queue lengths  $X = (X_1, X_2, \dots, X_N)$  and the total loss process  $L$  can be characterised as solutions of the following random time change equations

$$X_i(t) = X_i(0) + Y_{1,i} \left( \int_0^t \alpha_i(X(s), C(s)) ds \right) - Y_{-1,i} \left( \int_0^t \beta_i(X(s), C(s)) ds \right), \quad (6.3.3)$$

$$L(t) = L(0) + Y_0 \left( \int_0^t \gamma(X(s), C(s)) ds \right), \quad (6.3.4)$$

where  $Y_{1,i}$ ,  $Y_{-1,i}$  and  $Y_0$  are independent unit Poisson processes and the intensities  $\alpha_i, \beta_i, \gamma : \mathbb{N}_0^N \times \mathbb{E} \rightarrow \mathbb{R}_+$  are defined as follows

$$\alpha_i(x, c) = \begin{cases} \lambda_c \pi_i / \sum_{j \in [N]} \pi_j \mathbb{1}(x_j < K_j) & \text{if } x_i < K_i, \\ 0 & \text{otherwise,} \end{cases} \quad (6.3.5)$$

$$\beta_i(x, c) = \mu_{i,c} \mathbb{1}(0 < x_i \leq K_i), \quad (6.3.6)$$

$$\gamma(x, c) = \lambda_c \mathbb{1}(x = (K_1, K_2, \dots, K_N)). \quad (6.3.7)$$

Notice the explicit dependence of  $X$  and  $L$  on the modulating chain  $C$ . Also note that  $(X, L)$  is no longer Markovian, but  $(C, X, L)$  still is. In order to compute the probabilities of  $(C, X, L)$ , we solve the corresponding CMEs. Write  $p_t(c, x, y) := P(C(t) = c, X(t) = x, L(t) = y)$ , for  $c \in \mathbb{E}, x \in \{0, 1, \dots, K_1\} \times \{0, 1, \dots, K_2\} \times \dots \times \{0, 1, \dots, K_N\}$  and  $y \in \mathbb{N}_0$ . Then, we have

$$\begin{aligned} \frac{d}{dt} p_t(c, x, y) = & \sum_i \kappa_i(c - \eta_i) p_t(c - \eta_i, x, y) + \sum_j \alpha_j(x - e_j, c) p_t(c, x - e_j, y) \\ & + \sum_j \beta_j(x + e_j, c) p_t(c, x + e_j, y) + \gamma(x, c) p_t(c, x, y - 1) \\ & - \left( \sum_i \kappa_i(c) + \sum_j \alpha_j(x, c) + \sum_j \beta_j(x, c) + \gamma(x, c) \right) p_t(c, x, y). \end{aligned} \quad (6.3.8)$$

Note that the system of equations in (6.3.8) is infinite dimensional. However, as before, the probabilities of  $X$  are eventually obtained after marginalisation. Also, since the process  $C$  is not dependent on  $(X, L)$ , we can compute the probabilities of  $C$  separately (independent of  $(X, L)$ ). Indeed, we can solve

$$\frac{d}{dt} r_t(c) = \sum_i \kappa_i(c - \eta_i) r_t(c - \eta_i) + \sum_i \kappa_i(c) r_t(c), \quad (6.3.9)$$

with the initial condition  $r_0(c) = P(C(0) = c)$ , where we define  $r_t := P(C(t) = c)$ . However, obtaining the probabilities of  $C$  alone is not of much help for optimal scheduling, which we discuss next.

### 6.3.2 Optimal probabilistic scheduling

Our performance metric is the expected total loss  $E[L(t)]$ , which satisfies the following integral equation

$$E[L(t)] = E[L(0)] + \int_0^t \sum_{c \in \mathbb{E}} \lambda_c P(C(s) = c, X(s) = (K_1, K_2, \dots, K_N)) ds. \quad (6.3.10)$$

Now, we minimise  $E[L(T)]$  as a function of the scheduling probability vector  $\pi$ . Therefore, we define the optimal probabilistic schedule to be the optimiser of the total expected loss, i.e.,

$$\pi_{\text{opt}} := \underset{\pi}{\operatorname{argmin}} E[L(T)]. \quad (6.3.11)$$

Note that the optimal probabilistic schedule thus obtained is necessarily dependent on the properties of the modulating Markov chain  $C$ . Since the properties of the modulating chain are application-specific, the present approach promises application-specific optimal probabilistic scheduling.

**Example 6.3.1.** In many real-life applications, the environment under consideration is dichotomised. The dichotomy can arise, for instance, by virtue of the presence or absence of an exogenous factor. Markov-modulated on-off processes are a prime example. Heavy and low traffic regimes are often considered to arise as a consequence of an external modulation. In order to capture such a situation, we adopt a two-state Markov-modulated queueing system, *i.e.*,  $\mathbb{E} = \{1, 2\}$ .

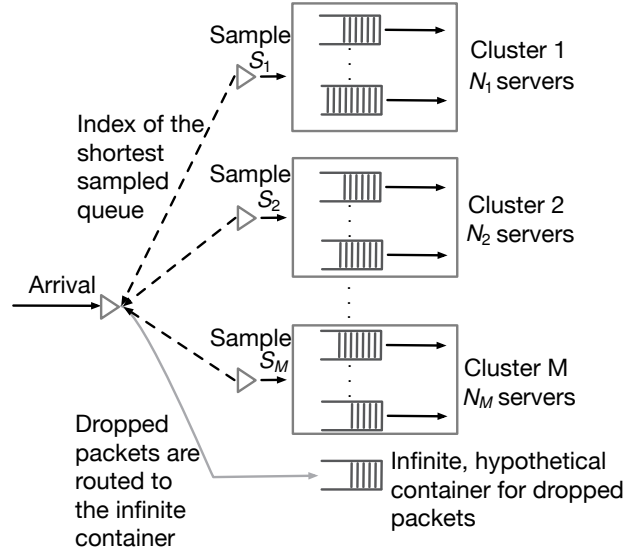
#### 6.4 A SCALING LIMIT: AN APPLICATION TO CLUSTERS OF SHARED SERVERS

As we discussed in Remark 6.2.1, the functions  $\alpha$ 's can be appropriately modified to reflect many other known scheduling algorithms. In this section, we shall provide a scaling limit of the system as the number of servers goes to infinity for a special class of CBQA scheduling algorithms, called the JMC schedules, which are a generalisation of randomised server selection schemes 1 and 2 considered in Mukhopadhyay, Karthik, and Mazumdar (2016) for infinite-buffer heterogeneous clusters of shared servers. In order to derive a scaling limit, we take the same basic modelling premise as Mukhopadhyay, Karthik, and Mazumdar (2016) and propose the following twofold generalisation: (i) we consider finite-buffers instead of infinite buffers; and (ii) we generalise the scheduling algorithm to allow user-defined cost functions associated with the queue-lengths of a randomly selected subset of servers. By virtue of the second generalisation, a wide range of innovative scheduling algorithms can be accommodated into the model and naturally, the schemes 1 and 2 in Mukhopadhyay, Karthik, and Mazumdar (2016) are obtained as special cases. We describe the queueing set-up in detail in the following.

##### 6.4.1 Description of the queueing set-up

We consider an  $N$ -server parallel processor sharing queueing system where the  $N$  servers are clubbed together into  $M$  ( $\ll N$ ) heterogeneous clusters of servers within each of which the servers are identical (see Figure 6.2). The cluster-structure could arise because of geographic location or otherwise. We assume the  $i$ -th cluster contains  $N_i$  identical servers with capacity  $\mu_i$ , the instantaneous rate at which each job is processed in the server. Naturally,  $N_1 + N_2 + \dots + N_M = N$  and  $M$  is assumed fixed throughout. Let  $I_i$  contain the indices of the servers in the  $i$ -th cluster. Then,  $|I_i| = N_i$  and  $\{I_1, I_2, \dots, I_M\}$  is a partition of  $[N]$ , *i.e.*,  $[N] = \bigcup_{i \in [M]} I_i$  and  $I_i$ 's are disjoint. For the sake of simplicity, let the buffer size of *all*  $N$  servers be  $K$ .

We assume jobs arrive at the system according to a Poisson process with rate  $\lambda_N$ . Each job is assumed to be of a random length that is exponentially distributed with mean  $m$ . The inter-arrival times, and the job lengths of each job are *all* assumed to be independent of each other. A local router is placed in each of the clusters. Upon arrival of a job, the main scheduler sends a request to all the  $M$  local routers. The local router in the  $i$ -th cluster randomly selects a subset of  $S_i$  servers and takes note of their queue lengths.



**Figure 6.2:** Schematic description of the JMC scheduling. There are  $M$  heterogeneous clusters of servers. Cluster  $i$  contains  $N_i$  servers, each of which is identical with capacity  $\mu_i$ . A local router is placed in each of the clusters. At the arrival of each job, the local router in cluster  $i$  randomly samples (with replacement)  $S_i$  servers and returns to the main scheduler the index of the shortest queue among the samples. The main scheduler then compares the costs associated with these servers. Finally, the incoming job is assigned to the server with minimum cost. If buffers are full, the jobs can not be assigned and therefore, are lost.

The local router then returns the index as well as the queue length of the server with the shortest queue length to the main scheduler. The main scheduler, having received  $M$  such queue lengths, computes their associated costs according to some user-specified cost function (which will be made precise later). Finally, the main scheduler chooses the server with the minimum cost and assigns the job to the chosen server. If the chosen server does not have adequate waiting room to accommodate the job, the job is lost. As before, the lost packets are assumed to be accumulated in a hypothetical container with infinite storage room. Finally each job leaves the system as soon as it receives its service.

#### 6.4.2 Join-Minimum-Cost scheduling

As described in Figure 6.2, at the arrival of each job, the main scheduler sends a request to each of the local routers. The local router in the  $i$ -th cluster samples  $S_i$  servers uniformly at random (with replacement). Suppose the queue lengths at the sampled servers at the  $i$ -th cluster are  $X_{j_1}, X_{j_2}, \dots, X_{j_l}$  with  $l = S_i$ . The local router returns to the main scheduler the index (and the queue length) of the server with the shortest queue length. That is, the local router computes

$$i_i := \operatorname{argmin}\{X_{j_k} \mid k \in [S_i]\}.$$

In case of a tie within a cluster, we break tie by choosing one of the tied indices uniformly at random. After receiving the indices  $\iota_1, \iota_2, \dots, \iota_M$  from all  $M$  clusters, the main scheduler assigns a cost to each of the corresponding queue lengths. Finally, comparing all the costs, the main scheduler assigns the job to the server with the minimum cost. Therefore, the index of the server to which the job is finally assigned is given by

$$\iota := \operatorname{argmin}\{\phi_k(X_{\iota_k}) \mid k \in [M]\},$$

where  $\phi_k$  is the cost function associated with the  $k$ -th cluster. We assume the cost functions are user-defined and continuous<sup>1</sup>. In case of a tie, we break tie by choosing one of the tied indices uniformly at random. Jobs leave the system as soon as their service has been provided. We assume the scheduling task (comparing the queues, computation of the costs and then comparing the costs) is instantaneous for modelling purposes. Having described the scheduling, we next attempt a scaling limit of the system as the number of servers within each cluster increases to infinity.

**Remark 6.4.1.** Notice that the above scheduling algorithm is indeed a generalisation of the two schemes considered in Mukhopadhyay, Karthik, and Mazumdar (2016). In particular, choosing  $\phi_k(x) = x$  corresponds to scheme 1 (a randomised version of the JSQ) and  $\phi_k(x) = x/\mu_k$ , to scheme 2. In our framework, other known variants of JSQ, or mixture of them, can be incorporated by choosing the cost functions appropriately. For instance, setting  $M = 1, S_1 = N, \phi_1(x) = x$  corresponds to the usual JSQ mechanism. Similarly,  $M = 1, S_1 = 2, \phi_1(x) = x$  corresponds to the power of 2 type JSQ.

### 6.4.3 A scaling limit

Before we present our scaling limit, we need to specify our technical assumptions. Therefore, we first define the key stochastic processes in the system. Let

$$Z_N(t) := \{Z_{n,i}^{(N)}(t) \mid i \in [M], n = 0, 1, 2, \dots, K\},$$

where  $Z_{n,i}^{(N)}(t)$  is the fraction of servers at the  $i$ -th cluster having at least  $n$  unfinished jobs (queue length in our parlance) at time  $t$ . That is,

$$Z_{n,i}^{(N)}(t) := \frac{1}{N_i} \sum_{k \in I_i} \mathbb{1}(X_k(t) \geq n).$$

We have deliberately suffixed the processes with  $N$  in order to emphasise their dependence on  $N$ . The process  $Z_N$  is a Markov process on the state space  $\times_{i \in [M]} \mathcal{Z}_i$ , the Cartesian product of  $\mathcal{Z}_1, \mathcal{Z}_2, \dots, \mathcal{Z}_M$  where

$$\mathcal{Z}_i := \{\{u_n\}_{n=0,1,2,\dots,K} \mid u_0 = 1, u_n \geq u_{n+1}, N_i u_n \in \mathbb{N}_0, \forall n = 0, 1, 2, \dots, K\}, \forall i \in [M].$$

<sup>1</sup> Note that continuity is vacuously satisfied if we restrict ourselves to integer-valued queues only, as we do in this work. However, continuity should be additionally assumed if we wish to generalise the cost function to other domains. In this work, we shall restrict ourselves to the integer-valued domain, for the sake of simplicity.

As we seek a scaling limit in this work, we also define the following space in which we expect the rows of the limiting process to lie

$$\mathcal{Z} := \{\{u_n\}_{n=0,1,2,\dots,K} \mid u_0 = 1, u_n \geq u_{n+1}, \forall n = 0, 1, 2, \dots, K\}.$$

That is, in the limit, we would expect  $Z_N$  to lie in  $\mathcal{Z}^M$ , the  $M$ -fold Cartesian product of  $\mathcal{Z}$  with itself. In order to metrize the space  $\mathcal{Z}^M$ , define the metric

$$\rho_{\mathcal{Z}^M}(u, v) := \sup_{i \in [M]} \sup_{n=0,1,2,\dots,K} \frac{|u_{n,i} - v_{n,i}|}{n+1}, \quad (6.4.1)$$

for  $u = (u^{(1)}, u^{(2)}, \dots, u^{(M)})$ ,  $v = (v^{(1)}, v^{(2)}, \dots, v^{(M)}) \in \mathcal{Z}^M$ , with  $u^{(i)} = (u_{n,i} \mid n = 0, 1, 2, \dots, K)$  and  $v^{(i)} = (v_{n,i} \mid n = 0, 1, 2, \dots, K)$  for each  $i \in [M]$ . Under  $\rho$ , the space  $\mathcal{Z}^M$  turns complete, separable and compact (Martin and Suhov 1999; Mukhopadhyay, Karthik, and Mazumdar 2016). Now, we lay down our technical assumptions.

**F1 (Arrival rate)** We assume the arrival rate grows linearly with  $N$ , *i.e.*,

$$\lim_{N \rightarrow \infty} \frac{\lambda_N}{N} = \lambda \in \mathbb{R}_+.$$

**F2 (Non-vanishing proportion of servers in each cluster)** Each cluster contains a non-vanishing proportion of the total pool of servers in the limit. That is, the cluster sizes grow linearly with  $N$ , *i.e.*

$$\lim_{N \rightarrow \infty} \frac{N_i}{N} = v_i \in \mathbb{R}_+, \quad \forall i \in [M].$$

**F3 (Decidability)** Given the index  $\iota$  (and the corresponding queue length) of the chosen server in accordance with the procedure JMC described in Section 6.4.2, the cost functions  $\phi_i$ 's allow us to decide the minimum of sampled queue lengths across the clusters. Suppose  $\iota \in I_i$ , *i.e.*, the chosen server is in the  $i$ -th cluster. Define

$$\theta_j(i, x) := \operatorname{argmin}_{y \in \{0,1,2,\dots,K\}} \{\phi_j(y) \geq \phi_i(x)\}, \quad \forall i, j \in [M], x \in \{0, 1, 2, \dots, K\}.$$

Our assumption of decidability amounts to demanding  $\theta_j(i, x) \neq 0$  for *at least one*  $j$ , for all  $i \in [M]$  and for all  $x > 0$ .

**Remark 6.4.2.** The assumption F3 is made to avoid trivialities and is not actually crucial. In order to make clear what we mean, consider a cost function that violates the decidability assumption. For instance, take  $\phi_i(x) = 0$  for all  $x \in \{0, 1, 2, \dots, K\}$ . Such a cost function is trivial, and renders the JMC scheduling algorithm completely random. In order to avoid such trivialities, we impose F3.

There are three primary approaches to proving convergence of Markov processes (Ethier and Kurtz 1986). First, an operator semigroup approach in which convergence of



the Markov process is achieved by proving convergence of certain semigroups. Second, martingale characterisation approach, which consists of characterising the Markov process and its limit as solutions of a certain martingale problem. Martingale convergence theorems are useful for this approach. In Chapter 8, we shall provide an FCLT the proof of which carries this flavour. The third approach makes use of characterisation of the Markov processes involving random time changes. In Chapter 7, we shall prove various QSSAs for a special class of queueing systems using this approach.

In this work, we shall adopt the operator semigroup approach (Ethier and Kurtz 1986) to prove convergence of the Markov process  $Z_N$ . Therefore, define a sequence of one-parameter families of operators  $\{T_N(t)\}_{t \in \mathcal{T}}$  as follows

$$T_N(t)f(z_0) := \mathbb{E}[f(Z_N(t)) \mid Z_N(0) = z_0], \quad \forall t \in \mathcal{T}, u_0 \in \times_{i \in [M]} \mathcal{Z}_i,$$

for continuous functions  $f : \times_{i \in [M]} \mathcal{Z}_i \rightarrow \mathbb{R}$ , and  $z_0 \in \times_{i \in [M]} \mathcal{Z}_i$ . The family  $\{T_N(t)\}_{t \in \mathcal{T}}$  defines a (contraction) semigroup by the Chapman-Kolmogorov property of Markov processes (Ethier and Kurtz 1986, Chapter 4). The convergence of this semigroup is proved by showing convergence of the corresponding sequence of (infinitesimal) generators. Therefore, define the generator  $A_N$  of the Markov process  $Z_N$  as

$$\begin{aligned} A_N f(u) &:= \lambda_N \sum_{i=1}^M \sum_{n=1}^K \left( (u_{n-1,i})^{S_i} - (u_{n,i})^{S_i} \right) \prod_{j \in [M] \setminus \{i\}} (u_{\theta_j(i,n-1),j})^{S_j} \left( f(u + \frac{1}{N_i} e_{n,i}) - f(u) \right) \\ &\quad + \sum_{i=1}^M \sum_{n=1}^K \frac{N_i \mu_i}{m} (u_{n,i} - u_{n+1,i}) \left( f(u - \frac{1}{N_i} e_{n,i}) - f(u) \right), \end{aligned} \quad (6.4.2)$$

where  $u = \{(u^{(1)}, u^{(2)}, \dots, u^{(M)}) \mid u^{(i)} = \{u_{n,i}\}_{n \in \{0,1,2,\dots,K\}} \in \mathcal{Z}_i, i \in [M]\} \in \times_{i \in [M]} \mathcal{Z}_i$  and  $e_{n,i} := \{(a_1, a_2, \dots, a_M) \mid a_j = \{\mathbb{1}(j = i, k = n)\}_{k \in \{0,1,2,\dots,K\}}, \forall j \in [M]\}$ . We set  $u_{K+1,i} = 0$  for all  $i \in [M]$ . Looking at the generator given in (6.4.2), we expect, at least intuitively, the limiting process  $z = \{z_{n,i} \mid i \in [M], n = 0, 1, 2, \dots, K\}$  to satisfy the integral equation

$$z(t) = z(0) + \int_0^t \mathbb{F}(z(s)) ds, \quad (6.4.3)$$

where the operator  $\mathbb{F}(z(s)) := \{\mathbb{F}_{n,i}(z(s)) \mid i \in [M], n = 0, 1, 2, \dots, K\}$  is given by

$$\begin{aligned} \mathbb{F}_{0,i}(u) &:= 0, \quad \forall i \in [M], \\ \mathbb{F}_{n,i}(u) &:= \frac{\lambda}{v_i} \left( (u_{n-1,i})^{S_i} - (u_{n,i})^{S_i} \right) \prod_{j \in [M] \setminus \{i\}} (u_{\theta_j(i,n-1),j})^{S_j} - \frac{\mu_i}{m} (u_{n,i} - u_{n+1,i}), \end{aligned} \quad (6.4.4)$$

for  $i \in [M]$  and  $n = 1, 2, \dots, K$ . We shall justify this intuition in the following. Before proceeding with the technical details, we discuss some properties of the proposed limit.

**Lemma 6.4.1.** *For any starting point  $u \in \mathcal{Z}^M$ , the solution to the integral equation (6.4.3) with the operators defined in (6.4.4) is unique on  $\mathcal{T}$ .*

The proof follows by Picard's iterative technique. For the sake of completeness, it is provided in Appendix D.2. As the limiting process  $z(t)$  depends on the initial value  $z(0)$ , we introduce the notation  $z(t, u)$  to denote the solution of the integral equation (6.4.3)

with  $z(0) = u$ . We require certain smoothness of the solution  $z(t, u)$  and bounded partial derivatives with respect to the initial point  $u$ . We check these conditions in Appendix D. Now, we find the limit of the generators.

**Remark 6.4.3** (Explanation for the generator). The generator defined in (6.4.2) is constructed by considering all the jumps of  $Z_N$ . The first part of (6.4.2) is due to an arrival of a customer (admittance, to be precise) and the second part corresponds to a departure of a customer after service. Consider the assignment of a job to a server with exactly  $n - 1$  unfinished jobs (queue length) in the  $i$ -th cluster when the system  $Z_N$  is in state  $u \in \times_{i \in [M]} \mathcal{Z}_i$ . This entails a jump from  $u$  to the state  $u + e_{n,i}/N_i$ . The term  $((u_{n-1,i})^{S_i} - (u_{n,i})^{S_i}) \prod_{j \in [M] \setminus \{i\}} (u_{\theta_j(i, n-1), j})^{S_j}$  gives the probability, under JMC scheduling, of a job to be assigned to a server with exactly  $n - 1$  unfinished jobs in the  $i$ -cluster. Under JMC scheduling, this happens only when the following two events happen.

1. At least one of the  $S_i$  sampled servers in the  $i$ -th cluster has exactly  $n - 1$  unfinished jobs and the others have at least  $n$  unfinished jobs.
2. In the light of F3, the fact that the main scheduler selects a server from the  $i$ -cluster implies that *all* the sampled servers in the  $j$ -th cluster must have at least  $\theta_j(i, n - 1)$  unfinished jobs, for all  $j \neq i$ .

Furthermore, the rate at which customers depart a server in the  $i$ -th cluster is  $N_i \mu_i (u_{n,i} - u_{n+1,i}) / m$ , which explains the second part of (6.4.2).

**Lemma 6.4.2** (Convergence of the generators). *Let  $\mathcal{C} := \mathcal{C}(\mathcal{Z}^M)$  denote the space of all real-valued continuous functions defined on  $\mathcal{Z}^M$ . Consider the subspace  $\mathcal{C}_D \subseteq \mathcal{C}$  of functions for which the partial derivatives*

$$\frac{\partial}{\partial u_{n,i}} z(t, u), \quad \frac{\partial^2}{\partial u_{n,i}^2} z(t, u), \quad \text{and} \quad \frac{\partial^2}{\partial u_{n,j} \partial u_{n,i}} z(t, u)$$

*exist for all  $u \in \mathcal{Z}^M$  and are uniformly bounded by some constant. Then, for all  $f \in \mathcal{C}_D$ ,*

$$\lim_{N \rightarrow \infty} A_N f(u) = \left. \frac{d}{dt} f(z(t, u)) \right|_{t=0}, \quad (6.4.5)$$

*where  $z$  is the solution of the integral equation (6.4.3).*

The proof of Lemma 6.4.2 is similar to the proof of Martin and Suhov (1999, Theorem 2). However, for the sake of completeness, it is provided in Appendix D.2. With the convergence of the generators in Lemma 6.4.2, we are now ready to prove convergence of the operator semigroup  $\{T_N(t)\}_{t \in \mathcal{T}}$ .

**Theorem 6.4.1** (Convergence of the operator semigroup). *Under the Join-Minimum-Cost scheduling algorithm, for any  $f \in \mathcal{C}_D$ , and  $t \in \mathcal{T}$ , we have the following convergence of the operator semigroup  $\{T_N(t)\}_{t \in \mathcal{T}}$ ,*

$$\lim_{N \rightarrow \infty} \sup_{u \in \times_{i \in [M]} \mathcal{Z}_i} |T_N(t)f(u) - T(t)f(u)| = 0, \quad (6.4.6)$$

where the limiting operator semigroup  $\{T(t)\}_{t \in \mathcal{T}}$  is defined by  $T(t)f(u) := f(z(t, u))$ , and is generated by the generator

$$Af(u) := \lim_{h \rightarrow 0+} \frac{T(t+h)f(u) - T(t)f(u)}{h} = \left. \frac{d}{dt} f(z(t, u)) \right|_{t=0},$$

where  $z$  is the solution of the integral equation (6.4.3).

*Proof of Theorem 6.4.1.* First note that the space  $\mathcal{C}_D$  is dense in  $\mathcal{C}$ . Also, both the semigroups  $\{T_N(t)\}_{t \in \mathcal{T}}$  and  $\{T(t)\}_{t \in \mathcal{T}}$  are strongly continuous and contracting (Ethier and Kurtz 1986). Moreover,  $\mathcal{C}_D$  is also a core of  $A$ . Therefore, following the same approach as Martin and Suhov (1999, Theorem 2), Mukhopadhyay, Karthik, and Mazumdar (2016), and by virtue of Lemma 6.4.2, we get the asserted convergence of the semigroups with the application of Ethier and Kurtz (1986, Chapter 1, Theorem 6.1).  $\square$

Having shown the convergence of the operator semigroup  $\{T_N(t)\}_{t \in \mathcal{T}}$  to  $\{T(t)\}_{t \in \mathcal{T}}$  in Theorem 6.4.1, the convergence of the Markov process  $Z_N$  follows immediately in the light of Ethier and Kurtz (1986, Chapter 4, Theorem 2.11), as also noted in Mukhopadhyay, Karthik, and Mazumdar (2016) also for the infinite-buffer case.

**Theorem 6.4.2** (Convergence of the proportions). *If  $\lim_{N \rightarrow \infty} Z_N(0) = z_0$ , for some non-random  $z_0 \in \mathcal{Z}^M$ , then*

$$Z_N \xrightarrow{\mathcal{D}} z, \text{ as } N \rightarrow \infty, \quad (6.4.7)$$

where the limiting process  $z$  is the solution to the integral equation (6.4.3) with  $z(0) = z_0$ , i.e.,  $z(t) \equiv z(t, z_0)$ , and lies in  $\mathcal{Z}^M$ . Weak convergence is understood in the sense of Billingsley (1999) and Ethier and Kurtz (1986).

**Remark 6.4.4.** Note that questions concerning stability of the queueing system does not arise in our context because the system is stable regardless of the arrival and service rates by virtue of finiteness of the buffers. However, the accumulated loss process is increasing because it has only positive jumps. Since we are only concerned with the transient behaviour of the queueing system in this chapter, we do not attempt to find the stationary queue lengths of this system.

**Remark 6.4.5** (Cost function as a tuning parameter). The explicit dependence of limiting process, an autonomous system of ODEs in this case, on the function  $\theta_j$  corresponding to a JMC scheduling strategy is worth noting. These explicit dependencies can be exploited in practical applications. That is, the cost functions themselves can be tuned to yield better performance in real applications.

#### 6.4.4 Optimal control

As already indicated in Remark 6.4.5, we can treat the cost functions  $\phi_i$ 's as *local control* variables and devise a global metric that encapsulates the performance of the whole

system. In order to make the idea precise, let us first define  $\phi := (\phi_1, \phi_2, \dots, \phi_M)$  to be a vector-valued control that determines the asymptotic behaviour (in the number of servers) of the queueing system via the  $\theta_j$ 's, as described in Theorem 6.4.2. Note that the  $\phi_i$ 's need not be of the same functional form, *i.e.*, we may enforce different cost functions for different clusters. Let  $z$  be the solution to the integral equation (6.4.3) with  $z(0) = z_0$ , *i.e.*,  $z(t) \equiv z(t, z_0)$ . Then, treating the operator  $\mathbb{F}(z(t))$  as a function of  $\phi$  as well, we can treat (6.4.3) as the state equation in classical optimal control framework. We can now define the *global* cost functional  $J$  as follows

$$J(\phi) := U(z_0, z(T), \phi, T) + \int_T V(z(s), \phi, s) ds, \quad (6.4.8)$$

where  $z$  is subject to the state equation (6.4.3) with  $z(0) = z_0$ ,  $U$  is the end-point cost, and  $V$  is the running cost. The choice of  $U$  and  $V$  are application-dependent. Our goal is to choose  $\phi$  in such a way that  $J$  is minimised. Therefore, the optimal control seeks to find  $\phi^*$  within a well defined product space of permissible local cost functions such that

$$\phi^* := \arg \min_{\phi} J(\phi). \quad (6.4.9)$$

Concrete applications of the optimal control approach will be explored in a future work.

In this work, the emphasis has been put on the expected total loss, which we show satisfies an integral equation involving the probability of the buffers being full. It would be interesting to explore performance metrics other than the expected total loss. While the expected total loss is convenient to analyse from a mathematical modelling perspective, it is worthwhile to explore how the optimal probabilistic schedule is functionally related with the performance metric. Also, the scaling limit presented in this work shows the explicit dependence on the cost functions via the  $\theta$  functions. Such dependencies can be exploited to design optimal schedules.

In the next chapter, we shall consider another special queueing system that closely resembles the MM enzyme-catalysed CRNs. We shall derive the various QSSAs in this context. We shall also discuss the relevance of this queueing set-up in the context of the collaborative uploading scenario described in Section 1.1.

## QUASI-STEADY STATE APPROXIMATIONS

In this chapter, we shall focus on a special kind of queueing system that resembles the MM enzyme-catalysed CRNs (Cornish-Bowden 2004; Hammes 2012; I. H. Segel 1975). The correspondence between the queueing system representing the collaborative uploading problem described in Section 1.1 and the MM enzyme-catalysed CRNs becomes clear via the following analogy: consider the substrates as customers in a queueing system and the enzymes as the servers (see Section 2.3 for a discussion on this). Therefore, in the context of the collaborative uploading scenario, the molecules of the substrate  $S$  can be thought of as the data chunks that need to be transported. The different paths are the servers, the free enzymes  $E$  in the MM enzyme-kinetic CRN. However, in this special case, we assume the different paths can carry only one data chunk at a time. Therefore, the substrate-enzyme complex  $C$  can be thought of as the paths that are currently transporting a data chunk, *i.e.*, the busy servers. The binding of a substrate molecule and a molecule of the enzyme refers to the assignment of a data chunk to a free path. This engenders creation of a busy server, a molecule of the substrate-enzyme complex  $C$ . The unbinding of  $C$  into a molecule of the substrate  $S$  and a free enzyme  $E$  refers to an unsuccessful attempt by the path to transport the data chunk. Naturally, the unbinding has the implication that the data chunk needs to be assigned once again, and the busy server turns into a free server. The products  $P$  are the data chunks that are *already* transported to the cloud. See Table 7.1. With these analogies, we can see that the MM enzyme-kinetic CRN describes the collaborative uploading scenario with the additional constraint that there is no room for queueing.

In this chapter, we shall make use of the random time change representation of the queueing system and derive various QSSAs. The QSSAs are particularly useful when the number of data chunks to transmit is too large compared to the number of paths available. Since the QSSAs are a popular tool in the physical chemistry literature, our derivations of the QSSAs are expectedly useful not only in queueing theoretic domain but also in physical chemistry. Therefore, in order to reach a wider audience, we shall adopt the MM description of the queueing system in the following and proceed to develop our results. This also serves to bridge the gap between the communities working in physical chemistry and queueing theory.

## 7.1 WHY QSSA?

In chemistry and biology, we often come across CRNs where one or more of the species exhibit a different intrinsic time scale and tend to reach an equilibrium state quicker than the others. The QSSA is a commonly used tool to simplify the description of the dynamics of such systems. In particular, QSSA has been widely applied to the important class of MM enzyme-kinetic CRNs.

Traditionally, the enzyme kinetics has been studied using systems of ODEs. The ODE approach allows one to analyse various aspects of the enzyme dynamics such as asymp-

CRN	Queueing theory	Uploading problem
$S$	Customers	The data chunks or packets to be transported.
$E$	Servers	The free paths.
$C$	The busy or occupied servers	The occupied paths, <i>i.e.</i> , the paths that are currently transporting a packet.
$P$	The served customers	The data chunks or packets already transported.
$S + E \rightarrow C$	The customer starts getting served.	Assignment of a packet to a free path. This turns the free path into a busy path.
$S + E \leftarrow C$	Service failure.	Unsuccessful attempt by the path to transport the data chunk. This has the implication that the data chunk needs to be assigned once again, and the busy server turns into a free server.
$C \rightarrow P + E$	Successful completion of service	Successful transport of a data chunk. Once the chunk is delivered, the busy server becomes free.

**Table 7.1:** Correspondence between Michaelis-Menten enzyme kinetics and the uploading problem.

otic stability. However, it ignores the fluctuations of the enzyme reaction network due to intrinsic noise and instead focuses on the averaged dynamics. If accounting for this intrinsic noise is required, the use of an alternative stochastic reaction network approach may be more appropriate, especially when some of the species have low copy numbers or when one is interested in predicting the molecular fluctuations of the system. It is well known that such molecular fluctuations in the species with small numbers, and stochasticity in general, can lead to interesting dynamics. For instance, in a recent paper Perez-Carrasco et al. (2016), the authors gave an account of how intrinsic noise controls and alters the dynamics, and the steady state of morphogen-controlled bistable genetic switches. In D. F. Anderson, Cappelletti, et al. (2017), the authors show that, in general, the behaviours of the deterministic system and the stochastic system can be vastly different with regards to the possibility of an explosion. In particular, they provide examples of an explosive stochastic system whose deterministic counterpart admits bounded solutions, and also non-explosive stochastic models whose deterministic counterpart suffers a blow-up. It is also worthwhile to note that there are stochastic reaction networks whose associated stochastic process explodes but the CME still admits a constant solution. Therefore, studying the behaviour of the deterministic model is generally inadequate. Stochastic models have been strongly advocated by many in recent literature (Assaf and Meerson 2017; Biancalani and Assaf 2015; Bressloff 2017; Bressloff and Newby 2013; Newby 2012, 2015). In this chapter, we consider such stochastic models in the context of QSSA and the MM enzyme kinetics and relate them to the deterministic ones that are well known from the chemical physics literature. In order to illustrate how the probabilistic tools can be used to derive various QSSAs for more general enzyme

kinetics than the MM reaction network, we also briefly consider a fully competitive ESI system in Appendix E.1.

The QSSAs are very useful from a practical perspective. They not only reduce the model complexity, but also allow us to better relate it to experimental measurements by averaging out the unobservable or difficult-to-measure species. A substantial body of work has been published to justify such QSSA reductions in deterministic models, typically by means of perturbation theory (Bersani and Dell’Acqua 2011; Dingee and Anton 2008; Schneider and Wilhelm 2000; Schnell and Mendoza 1997; L. A. Segel and Slemrod 1989; Stiefenhofer 1998). In contrast to this approach, we derive the QSSA reductions using stochastic multi-scaling techniques (Ball et al. 2006; Kang and Kurtz 2013). Although our approach is applicable more generally, we focus below on the three well established enzyme-kinetic QSSAs, namely the standard QSSA (sQSSA), the total QSSA (tQSSA), and the reversible QSSA (rQSSA) for the MM enzyme kinetics. We also briefly consider a fully competitive ESI system and show how sQSSA and tQSSA can be derived based on our multi-scaling techniques. We show that these QSSAs are a consequence of the (Poisson) law of large numbers for the stochastic reaction network under different scaling regimes. A similar approach has been recently taken in J. K. Kim, Rempała, and Kang (2017) with respect to a particular type of QSSA (tQSSA, see below in Section 7.2). However, our current derivation is different in that it entirely avoids a spatial averaging argument used in J. K. Kim, Rempała, and Kang (2017). Such an argument requires additional assumptions that are difficult to verify in practice.

## 7.2 QSSAS FOR DETERMINISTIC MICHAELIS-MENTEN KINETICS

The MM enzyme-catalysed reaction networks have been studied in depth over past several decades (Cornish-Bowden 2004; Hammes 2012; I. H. Segel 1975) and have been described in various forms. Although the methods discussed below certainly apply to more general networks of reactions describing enzyme kinetics, we adopt the simplest (and minimal) description for illustration purpose. In its simplest form, the MM enzyme-catalysed network of reactions describes reversible binding of a free enzyme ( $E$ ) and a substrate ( $S$ ) into an enzyme-substrate complex ( $C$ ), and irreversible conversion of the complex  $C$  to the product ( $P$ ) and the free enzyme  $E$  (see also Section 2.3). The enzyme-catalysed reactions are schematically described as



where  $k_1$  and  $k_{-1}$  are the reaction rate constants for the reversible enzyme binding in the units of  $M^{-1}s^{-1}$  and  $s^{-1}$  while  $k_2$  is the rate constant for the product creation in



the unit of  $s^{-1}$ . Applying the law of mass-action to (7.2.1), temporal changes of the concentrations are described by the following system of ODEs

$$\begin{aligned}\frac{d}{dt}[S] &= -k_1[S][E] + k_{-1}[C], \\ \frac{d}{dt}[E] &= -k_1[S][E] + k_{-1}[C] + k_2[C], \\ \frac{d}{dt}[C] &= k_1[S][E] - k_{-1}[C] - k_2[C], \\ \frac{d}{dt}[P] &= k_2[C],\end{aligned}\tag{7.2.2}$$

where the bracket notation  $[\cdot]$  refers to the concentration of species. In a closed system, there are two conservation laws for the total amount of enzyme and substrate

$$[E_0] := [E] + [C], \quad [S_0] := [S] + [C] + [P].\tag{7.2.3}$$

These conservation laws not only reduce (7.2.2) to two equations, but also play an important role in the analysis of the reaction network given in (7.2.1). It is worth mentioning that some authors also consider an additional reversible reaction in the form of binding of the product  $P$  and the free enzyme  $E$  to produce the enzyme-substrate complex  $C$ , *i.e.*,  $P + E \rightleftharpoons C$ . We remark that should we expand the model in (7.2.1) to include such a reaction, our discussion in later sections would remain largely the same requiring only simple modifications.

Leonor Michaelis and Maud Menten investigated the enzymatic kinetics in (7.2.1) and proposed a mathematical model for it in Michaelis and Menten (1913). They suggested an approximate solution for the initial velocity of the enzyme inversion reaction in terms of the substrate concentrations. Following their work, numerous attempts have been made to obtain approximate solutions of (7.2.2) under various quasi-steady-state assumptions. Several conditions on the rate constants have also been proposed for the validity of such approximations. For example, Briggs and Haldane mathematically derived the MM equation, which is now known as sQSSA (Briggs and Haldane 1925). The sQSSA is based on the assumption that the complex reaches its steady state quickly after a transient time, *i.e.*,  $\frac{d}{dt}[C] \approx 0$  (L. A. Segel and Slemrod 1989). This approximation is found to be inaccurate when the enzyme concentration is not small compared to that of the substrate. The condition for the validity of the sQSSA was first suggested as  $[E_0] \ll [S_0]$  by Laidler (Laidler 1955), and a more general condition was derived as  $[E_0] \ll [S_0] + K_M$  by L. A. Segel (1988) and L. A. Segel and Slemrod (1989), where  $K_M := (k_2 + k_{-1})/k_1$  is the so-called MM constant.

Borghans et al. later extended the sQSSA to the case with an excessive amount of enzyme and derived the tQSSA by introducing a new variable for total substrate concentration (Borghans, De Boer, and L. A. Segel 1996). In the tQSSA, one assumes that the total substrate concentration changes on a slow time scale and that the complex reaches its steady state quickly after a transient time,  $\frac{d}{dt}[C] \approx 0$ . Then, the complex concentration  $[C]$  is found as a solution of a quadratic equation. Approximating  $[C]$  in a simple way, they proposed a necessary and sufficient condition for the validity of tQSSA as

$$([E_0] + [S_0] + K_M)^2 \gg K[E_0],\tag{7.2.4}$$



where  $K = k_2/k_1$  is the so-called Van Slyke-Cullen constant (Van Slyke and Cullen 1914). Later, Tzafiri (2003) revisited the tQSSA and derived another set of sufficient conditions for the validity of the tQSSA as  $\epsilon := (K/(2[S_0])) f(r([S_0])) \ll 1$  where  $f(r) = (1 - r)^{-1/2} - 1$  and  $r([S_0]) = 4[E_0][S_0]/([E_0] + [S_0] + K_M)^2$ . He argued that this sufficient condition was always roughly satisfied by showing  $\epsilon$  was less than 1/4 for all values of  $[E_0]$  and  $[S_0]$ . The tQSSA was later improved by Dell'Acqua and Bersani (2012) at high enzyme concentrations when (7.2.4) is satisfied.

The rQSSA was first suggested as an alternative to the sQSSA by L. A. Segel and Slemrod (1989). In the rQSSA, the substrate, instead of the complex, was assumed to be at steady state,  $\frac{d}{dt}[S] \approx 0$ , and the domain of the validity of the rQSSA was suggested as  $[E_0] \gg K$ . Then, Schnell and Maini showed that at high enzyme concentration, the assumption  $\frac{d}{dt}[S] \approx 0$  was more appropriate in the rQSSA than the assumption  $\frac{d}{dt}[C] \approx 0$  used in the sQSSA or tQSSA due to possibly large error during the initial stage of the reactions (Schnell and Maini 2000). They derived necessary conditions for the validity of the rQSSA as  $[E_0] \gg K$  and  $[E_0] \gg [S_0]$ . In the following sections, we will provide alternative derivations of these different conditions.

### 7.3 MULTI-SCALE STOCHASTIC MICHAELIS-MENTEN KINETICS

Let  $X_S$ ,  $X_E$ ,  $X_C$ , and  $X_P$  denote the copy numbers of molecules of the substrates  $S$ , the enzymes  $E$ , the enzyme-substrate complex  $C$ , and the product  $P$  respectively. We assume the evolution of these copy numbers is governed by a Markovian dynamics given by the following stochastic equations

$$\begin{aligned} X_S(t) &= X_S(0) - Y_1 \left( \int_0^t \kappa'_1 X_S(s) X_E(s) ds \right) + Y_{-1} \left( \int_0^t \kappa'_{-1} X_C(s) ds \right), \\ X_E(t) &= X_E(0) - Y_1 \left( \int_0^t \kappa'_1 X_S(s) X_E(s) ds \right) + Y_{-1} \left( \int_0^t \kappa'_{-1} X_C(s) ds \right) + Y_2 \left( \int_0^t \kappa'_2 X_C(s) ds \right), \\ X_C(t) &= X_C(0) + Y_1 \left( \int_0^t \kappa'_1 X_S(s) X_E(s) ds \right) - Y_{-1} \left( \int_0^t \kappa'_{-1} X_C(s) ds \right) - Y_2 \left( \int_0^t \kappa'_2 X_C(s) ds \right), \\ X_P(t) &= X_P(0) + Y_2 \left( \int_0^t \kappa'_2 X_C(s) ds \right), \end{aligned} \tag{7.3.1}$$

where  $Y_1, Y_{-1}$  and  $Y_2$  are independent unit Poisson processes and  $t \geq 0$ . The quantities  $\kappa'_1, \kappa'_{-1}, \kappa'_2$  are the stochastic reaction rate constants. They can be related to the deterministic reaction rate constants by means of the Avogadro's number. We shall make this point precise in Section 7.4. We denote  $X_{E_0} := X_E(t) + X_C(t)$  and  $X_{S_0} := X_S(t) + X_C(t) + X_P(t)$ , and as in the deterministic model (7.2.2) in previous section assume that the total substrate and enzymes copy numbers,  $X_{S_0}$  and  $X_{E_0}$ , are conserved in time. As shown in Ball et al. (2006) and Kang and Kurtz (2013), the representation (7.3.1) is especially helpful in analysing systems with multiple time scales or involving species with abundances varying over different orders of magnitude. Unlike the CMEs, (7.3.1) explicitly reveals the relations between the species abundances and the reaction rates.

In the reaction system (7.2.1), various scales can exist in the species numbers and reaction rate constants, which determine time scales of the species involved. In order to

relate these scales, we first define a scaling parameter  $N$  to express the orders of magnitude of species copy numbers and rate constants as powers of  $N$ . We note that  $1/N$  plays a similar role as the expansion parameter (usually denoted by  $\epsilon$ ) in the singular perturbation analysis of deterministic models (L. A. Segel and Slemrod 1989). Denoting scaling exponents for the species  $i$  and the  $k$ -th rate constant by  $\alpha_i$  and  $\beta_k$  respectively, we express unscaled species copy numbers and rate constants as some powers of  $N$  as

$$X_i(t) = N^{\alpha_i} Z_i^N(t), \text{ for } i = S, E, C, P \quad \text{and} \quad \kappa'_k = N^{\beta_k} \kappa_k, \text{ for } k = 1, -1, 2, \quad (7.3.2)$$

so that the scaled variables and constants,  $Z_i^N(t)$  and  $\kappa_k$ , are approximately of order 1 (denoted as  $O(1)$ ). In  $Z_i^N$ , the superscript represents the dependence of the scaled species numbers on  $N$ . To express different time scales as powers of  $N$ , we apply a time change by replacing  $t$  with  $N^\gamma t$ . The scaled species number after the time change is given by

$$X_i(N^\gamma t) = N^{\alpha_i} Z_i^N(N^\gamma t) = N^{\alpha_i} Z_i^{N,\gamma}(t).$$

Therefore,  $\{Z^{N,\gamma}\} := \left\{ \left( Z_S^{N,\gamma}, Z_E^{N,\gamma}, Z_C^{N,\gamma}, Z_P^{N,\gamma} \right) \right\}$  becomes a parametrised family of stochastic processes satisfying

$$\begin{aligned} Z_S^{N,\gamma}(t) &= Z_S^N(0) + N^{-\alpha_S} \left[ -Y_1 \left( \int_0^t N^{\rho_1+\gamma} \kappa_1 Z_S^{N,\gamma}(s) Z_E^{N,\gamma}(s) ds \right) \right. \\ &\quad \left. + Y_{-1} \left( \int_0^t N^{\rho_{-1}+\gamma} \kappa_{-1} Z_C^{N,\gamma}(s) ds \right) \right], \\ Z_E^{N,\gamma}(t) &= Z_E^N(0) + N^{-\alpha_E} \left[ -Y_1 \left( \int_0^t N^{\rho_1+\gamma} \kappa_1 Z_S^{N,\gamma}(s) Z_E^{N,\gamma}(s) ds \right) \right. \\ &\quad \left. + Y_{-1} \left( \int_0^t N^{\rho_{-1}+\gamma} \kappa_{-1} Z_C^{N,\gamma}(s) ds \right) + Y_2 \left( \int_0^t N^{\rho_2+\gamma} \kappa_2 Z_C^{N,\gamma}(s) ds \right) \right], \quad (7.3.3) \\ Z_C^{N,\gamma}(t) &= Z_C^N(0) + N^{-\alpha_C} \left[ Y_1 \left( \int_0^t N^{\rho_1+\gamma} \kappa_1 Z_S^{N,\gamma}(s) Z_E^{N,\gamma}(s) ds \right) \right. \\ &\quad \left. - Y_{-1} \left( \int_0^t N^{\rho_{-1}+\gamma} \kappa_{-1} Z_C^{N,\gamma}(s) ds \right) - Y_2 \left( \int_0^t N^{\rho_2+\gamma} \kappa_2 Z_C^{N,\gamma}(s) ds \right) \right], \\ Z_P^{N,\gamma}(t) &= Z_P^N(0) + N^{-\alpha_P} Y_2 \left( \int_0^t N^{\rho_2+\gamma} \kappa_2 Z_C^{N,\gamma}(s) ds \right), \end{aligned}$$

where  $\rho_1 := \alpha_S + \alpha_E + \beta_1$ ,  $\rho_{-1} := \alpha_C + \beta_{-1}$ , and  $\rho_2 := \alpha_C + \beta_2$ . As seen from (7.3.3), the values of  $\rho$ 's,  $\alpha$ 's and  $\gamma$ 's determine the temporal dynamics of the scaled random processes. For example, consider the limiting behaviour of the scaled process for the first reaction in the equation for  $S$ ,

$$N^{-\alpha_S} Y_1 \left( \int_0^t N^{\rho_1+\gamma} \kappa_1 Z_S^{N,\gamma}(s) Z_E^{N,\gamma}(s) ds \right). \quad (7.3.4)$$

Assuming that  $Z_S^{N,\gamma}$  and  $Z_E^{N,\gamma}$  are  $O(1)$  in the time scale of interest, the limiting behaviour of the scaled process depends upon  $\rho_1$ ,  $\alpha_S$ , and  $\gamma$ . If the  $\rho_1 + \gamma < \alpha_S$ , the scaled process converges to zero as  $N$  goes to infinity. This means that the number of occurrences of the first reaction is outweighed by the order of magnitude of the species copy

number for  $S$ . When  $\rho_1 + \gamma = \alpha_S$ , the number of occurrences of the first reaction is comparable to the order of magnitude of the species copy number for  $S$ . Then, using the law of large numbers for the Poisson processes<sup>1</sup>, the limiting behaviour of (7.3.4) is approximately the same as that of

$$\int_0^t \kappa_1 Z_S^{N,\gamma}(s) Z_E^{N,\gamma}(s) ds. \quad (7.3.5)$$

Lastly, when  $\rho_1 + \gamma > \alpha_S$ , the first reaction occurs so frequently that the scaled process in (7.3.4) tends to infinity. The limiting behaviours of other scaled processes are determined similarly. Using the scaled processes involving the reactions where  $S$  is produced or consumed, we can choose  $\gamma$  so that  $Z_S^{N,\gamma}(t)$  becomes  $O(1)$ . Therefore, we have  $\alpha_S = \max(\rho_1 + \gamma, \rho_{-1} + \gamma)$ , and the *time scale* of  $S$  is given by

$$\gamma = \alpha_S - \max(\rho_1, \rho_{-1}). \quad (7.3.6)$$

Therefore, the time scales of the species numbers and their limiting behaviours are decided by the scaling exponents for species numbers and reactions, that is, they are dictated by the choice of  $\alpha$ 's and  $\beta$ 's.

In order to prevent the system from vanishing to zero or exploding to infinity in the scaling limit, the parameters  $\alpha$ 's and  $\beta$ 's must satisfy what are known as the balance conditions (Kang and Kurtz 2013). Essentially, these conditions ensure that the scaling limit is  $O(1)$ . Intuitively, the largest order of magnitude of the production of species  $i$  should be the same as that of consumption of species  $i$ . For instance, in the MM reaction network described in Section 7.2, balance for the substrate  $S$  can be achieved in two ways. First, through the equation  $\rho_1 = \rho_{-1}$ , which *balances* the binding and unbinding of the enzyme to the substrate; and second, by making  $\alpha_S$  large enough so that the imbalance between the occurrences of the reversible binding of the enzyme to substrate can be nullified. This gives a restriction on the time scale  $\gamma$  as  $\gamma + \max(\rho_1, \rho_{-1}) \leq \alpha_S$ . Combining the equality and inequality for each species, we get species balance conditions as

$$\begin{aligned} \rho_1 = \rho_{-1} & \quad \text{or} \quad \gamma \leq \alpha_S - \max(\rho_1, \rho_{-1}), \\ \rho_1 = \max(\rho_{-1}, \rho_2) & \quad \text{or} \quad \gamma \leq \alpha_E - \max(\rho_1, \rho_{-1}, \rho_2), \\ \rho_1 = \max(\rho_{-1}, \rho_2) & \quad \text{or} \quad \gamma \leq \alpha_C - \max(\rho_1, \rho_{-1}, \rho_2), \\ \rho_2 + \gamma = 0 & \quad \text{or} \quad \gamma \leq \alpha_P - \rho_2. \end{aligned} \quad (7.3.7)$$

Even if the conditions in (7.3.7) are satisfied, additional conditions are often required to make the scaled species numbers asymptotically  $O(1)$ . For each linear combination of species, the collective production and consumption rates should be balanced. Otherwise, the time scale of the new variable consisting of the linear combination of the scaled species will be restricted up to some time. The additional conditions are

$$\begin{aligned} \rho_2 + \gamma = 0 & \quad \text{or} \quad \gamma \leq \max(\alpha_S, \alpha_C) - \rho_2, \\ \rho_1 = \rho_{-1} & \quad \text{or} \quad \gamma \leq \max(\alpha_C, \alpha_P) - \max(\rho_1, \rho_{-1}), \end{aligned} \quad (7.3.8)$$

<sup>1</sup> The strong law of large numbers states that, for a unit Poisson process  $Y$ ,  $\frac{1}{N}Y(Nu) \rightarrow u$  almost surely as  $N \rightarrow \infty$ , (see Ethier and Kurtz (1986)).

obtained by comparing collective production and consumption rates of  $S + C$  and  $C + P$ , respectively.

The multi-scaling technique allows one to produce a wide range of approximations by tuning the scaling exponents suitably to reflect different regimes of time-scale separation and species abundance. While the main purpose of this work is to show how the sQSSA, the tQSSA and the rQSSA can be derived directly from the stochastic description by means of an appropriate choice of the scaling exponents, several other trivial as well as nontrivial approximations, which have a quasi-steady state flavour, can be obtained using this technique. In fact, even for similar species abundance regimes, quite different limiting dynamics can be obtained from the stochastic system directly by virtue of the time-scale separation. This ability to engender a wide range of interesting limiting dynamics makes the multi-scaling technique a powerful tool for studying chemical reactions in general and enzyme kinetics, in particular. Therefore, before providing our main results, we present a simple example here.

**Example 7.3.1.** Consider the MM kinetics with the enzyme  $E$  in much greater abundance compared to the other species. We assume that initially enzyme and substrate amounts are non-zero while the initial copy numbers of enzyme-substrate complex and product are zero. We also assume that all reactions occur at rates in the same order of magnitude. In order to model such a pathological case in the deterministic setting, one could assume the enzyme concentration does not change over time, *i.e.*,  $\frac{d}{dt}[E] \approx 0$  and consider the following reduced model

$$\frac{d}{dt}[S] = -\tilde{k}_1[S] + k_{-1}[C], \quad \text{and} \quad \frac{d}{dt}[C] = \tilde{k}_1[S] - (k_{-1} + k_2)[C],$$

where we have absorbed the constant enzyme concentration into  $\tilde{k}_1$ . The above system qualitatively predicts rapid decay in substrate concentration and an initial growth in complex concentration because  $\tilde{k}_1$  is much greater than the other two reaction rate constants. Note that, since the initial copy number for the complex is zero, there will be stochastic fluctuations, at least initially (depending on the magnitudes of the reaction rate constants). Moreover, such a qualitative prediction does not provide insights into the inherent time scales of the different species.

Now, capturing this corner case in the stochastic framework, we set  $\alpha_S = \alpha_C = \alpha_P = 1$ ,  $\alpha_E = 2$ ,  $\beta_1 = -2$ , and  $\beta_{-1} = \beta_2 = 0$  based on our assumptions. Note that the order of magnitude of all propensities is the same as  $\rho_1 = \rho_{-1} = \rho_2 = 1$ . The chosen set of  $\alpha$ 's and  $\rho$ 's satisfies all balance equations in (7.3.7)-(7.3.8) but  $\rho_2 + \gamma = 0$ . The time scales of  $S$ ,  $C$ , and  $P$  are identified as  $\gamma = 0$  as defined in (7.3.6) while the time scale of  $E$  is given by  $\gamma = 1$ . Following the multi-scale technique described above, we obtain  $(Z_S^0, Z_C^0, Z_P^0)$  as a scaling limit of  $(Z_S^{N,0}, Z_C^{N,0}, Z_P^{N,0})$  as  $N$  goes to infinity. In particular, considering the time scale of  $S$ , and  $C$ , we get the following limiting equations

$$\begin{aligned} Z_S^0(t) &= Z_S(0) + \int_0^t \left( -\kappa_1 Z_S^0(s) Z_E(0) + \kappa_{-1} Z_C^0(s) \right) ds, \\ Z_C^0(t) &= Z_C(0) + \int_0^t \left( \kappa_1 Z_S^0(s) Z_E(0) - (\kappa_{-1} + \kappa_2) Z_C^0(s) \right) ds, \end{aligned} \tag{7.3.9}$$

where  $E$  is approximated as its initial value since the time scale of  $E$  is later than  $\gamma = 0$ . More interestingly, in the time scale of  $E$ , *i.e.*, when  $\gamma = 1$ , the averaged behaviours of  $S$ ,  $C$  and  $P$  are approximated by

$$\overline{Z_S^1}(t) = \overline{Z_C^1}(t) = 0, \quad \overline{Z_P^1}(t) = Z_S(0) + Z_C(0) + Z_P(0), \quad (7.3.10)$$

while  $Z_E^{N,1}$  converges to  $Z_E^1$ , which, expectedly, satisfies  $Z_E^1(t) = Z_E(0)$ . That is, in the time scale of the enzyme, there is no dynamic behaviour (time evolution) at all in the limit. Note that (7.3.10) is independent of the reaction rate constants. One can obtain such a behaviour from the deterministic system by *additionally* assuming  $\frac{d}{dt}[S] \approx 0$  and  $\frac{d}{dt}[C] \approx 0$ , which renders the ODE system completely trivial and our assumptions about the initial species abundances irrelevant. On the other hand, the averaged behaviour (7.3.10) is a direct implication of the multi-scale approximation in the time scale of the enzyme, rather than an additional assumption. Therefore, the multi-scale approximation tool allows us to study different behaviours (often in different time scales) directly from the stochastic description without making additional assumptions. In the following sections, we exploit this tool to derive the sQSSA, the tQSSA and the rQSSA for the stochastic MM kinetics (7.3.1).

#### 7.4 STANDARD QUASI-STEADY-STATE APPROXIMATION

In the deterministic sQSSA, one assumes that the substrate-enzyme complex  $C$  reaches its steady-state quickly after a brief transient phase while the other species are still in their transient states. Therefore, by setting  $\frac{d}{dt}[C] \approx 0$ , one approximates the steady state concentration of the complex. The steady state equation of the complex in (7.2.2) and the conservation of the total enzyme concentration in (7.2.3) give

$$[C] = \frac{[E_0][S]}{K_M + [S]}, \quad (7.4.1)$$

where  $K_M = (k_{-1} + k_2)/k_1$ . The substrate concentration is then given by

$$\frac{d}{dt}[S] = -\frac{k_2[E_0][S]}{K_M + [S]}. \quad (7.4.2)$$

The corresponding equations for  $[E]$  and  $[P]$  can be written similarly. This approximation is known as the sQSSA of the MM kinetics (7.2.1) under the deterministic setting.

Now, we use stochastic equations for the species copy numbers in (7.3.1) and apply the multi-scale approximation to derive an analogue of (7.4.1)-(7.4.2). Equations like (7.4.2) have been previously derived from the stochastic reaction network (Darden 1979, 1982). It was also revisited specifically using the multi-scale approximation method in D. F. Anderson and Kurtz (2011) and Kang and Kurtz (2013). However, for the sake of completeness, we furnish a brief description below. Assuming that  $E$  and  $C$  are on the faster time scale than  $S$  and  $P$ , consider the scaled processes in (7.3.3) with the following scaling exponents

$$\alpha_S = \alpha_P = 1, \quad \alpha_E = \alpha_C = 0, \quad \beta_1 = 0, \quad \beta_{-1} = \beta_2 = 1, \quad (7.4.3)$$

that is,  $\rho_1 = \alpha_S + \alpha_E + \beta_1 = 1$ ,  $\rho_{-1} = \alpha_C + \beta_{-1} = 1$ , and  $\rho_2 = \alpha_C + \beta_2 = 1$ . Note that when  $\gamma = 0$ , the above corresponds to assuming the abundances of the substrate and the product are order  $N$  while those of the enzyme and the enzyme-substrate complex are order 1. We are interested in the time scale of  $S$  given in (7.3.6). Plugging in the scaling exponent values in (7.4.3), the time scale of  $S$  we are interested in corresponds to  $\gamma = 0$ . Setting  $\gamma = 0$  in the scaled stochastic equations in (7.3.3) and writing  $Z_i^N$  instead of  $Z_i^{N,\gamma}$  for  $i = S, E, C, P$  one obtains from (7.4.3). Define  $M := Z_E^N(t) + Z_C^N(t)$  and

$$Z_C^N(t) := \int_0^t Z_C^N(s) ds = Mt - \int_0^t Z_E^N(s) ds.$$

Note that  $M = Z_E^N(0) + Z_C^N(0) = X_E(0) + X_C(0)$ , and that  $M$  does not depend on the scaling parameter  $N$ . As done in D. F. Anderson and Kurtz (2011) and Kang and Kurtz (2013), assume that  $Z_S^N(0) \rightarrow Z_S(0)$ . The scaled variables  $Z_S^N$  and  $Z_C^N$  are bounded so they are relatively compact in the finite time interval  $[0, \mathcal{T}]$ , where  $0 < \mathcal{T} < \infty$ . Then,  $(Z_S^N, Z_C^N)$  converges to  $(Z_S, Z_C)$  as  $N \rightarrow \infty$  and satisfies for every  $t > 0$ ,

$$\begin{aligned} Z_S(t) &= Z_S(0) - \int_0^t \kappa_1 Z_S(s) (M - \dot{Z}_C(s)) ds + \int_0^t \kappa_{-1} \dot{Z}_C(s) ds, \\ 0 &= \int_0^t \kappa_1 Z_S(s) (M - \dot{Z}_C(s)) ds - \int_0^t (\kappa_{-1} + \kappa_2) \dot{Z}_C(s) ds. \end{aligned} \quad (7.4.4)$$

Note that we get (7.4.4) by dividing the equation for  $Z_C^N(t)$  in (7.3.3) by  $N$  and taking the limit as  $N \rightarrow \infty$ . From (7.4.4), we get

$$\dot{Z}_S(t) = -\frac{\kappa_2 M Z_S(t)}{\kappa_M + Z_S(t)}, \quad \dot{Z}_C(t) = \frac{M Z_S(t)}{\kappa_M + Z_S(t)}, \quad (7.4.5)$$

where  $\kappa_M = (\kappa_{-1} + \kappa_2)/\kappa_1$ , which is precisely the sQSSA.

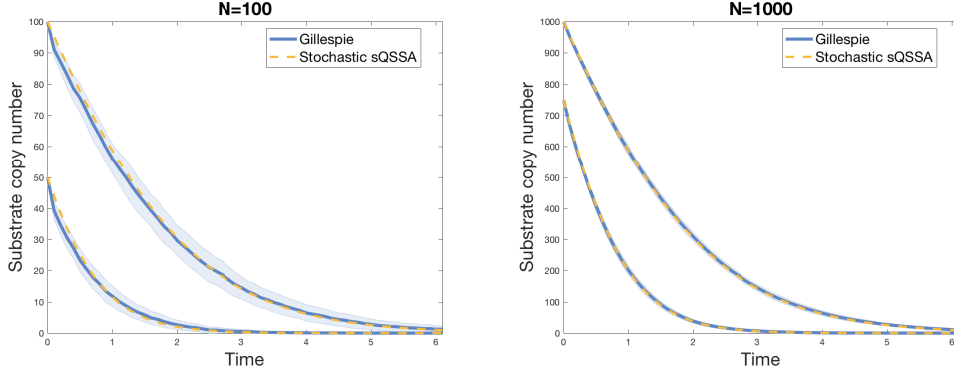
Note that we only use the Poisson law of large numbers and the conservation law to derive (7.4.5). In Figure 7.1, we compare the limit  $Z_S(t)$  in (7.4.5) with the scaled substrate copy number  $Z_S^N(t)$  in (7.3.3), obtained from 1000 realizations of the stochastic simulation using Gillespie's algorithm (Gillespie 1977). Figure 7.1 shows the agreement between the scaled process  $Z_S^N(t)$  and its limit  $Z_S(t)$ .

**CONDITIONS FOR SQSSA IN THE DETERMINISTIC SYSTEM** We have shown that the scaling exponents (7.4.3) indeed yielded the sQSSA. We now show how the conditions (7.4.3) are related to the conditions proposed in the literature for the validity of the deterministic sQSSA. First, we consider a general condition derived by L. A. Segel (1988) and L. A. Segel and Slemrod (1989),

$$[E_0] \ll [S_0] + K_M, \quad (7.4.6)$$

where  $K_M = (k_{-1} + k_2)/k_1$  is the MM constant. We rewrite (7.4.6) in terms of the species copy numbers and the stochastic reaction rate constants. The stochastic and the deterministic reaction rates are related as

$$(k_1, k_{-1}, k_2) = (V\kappa'_1, \kappa'_{-1}, \kappa'_2), \quad (7.4.7)$$



**Figure 7.1:** MM kinetics with sQSSA. The scaling limit of the substrate copy number, drawn in yellow dotted line, is compared with the mean substrate copy number, obtained from simulations using the Gillespie's algorithm and shown in blue. The light blue shaded region represents one standard deviation from the mean. Different choices of initial conditions are used to reflect the fact that convergence can be achieved under varying values of the conservation constant  $M$ . Simulation settings: **(a)**  $N = 100$ ,  $(X_S^N(0), X_E^N(0), X_C^N(0), X_P^N(0)) = (100, 10, 0, 0)$  for the upper curve and  $(50, 20, 0, 0)$  for the lower curve; and **(b)**  $N = 1000$ ,  $(X_S^N(0), X_E^N(0), X_C^N(0), X_P^N(0)) = (1000, 10, 0, 0)$  for the upper curve and  $(750, 20, 0, 0)$  for the lower curve. The reaction rate constants are  $(\kappa_1, \kappa_{-1}, \kappa_2) = (1, 1, 0.1)$  in both **(a)** and **(b)**.

where  $V$  is the system volume multiplied by the Avogadro's number (Kurtz 1972). We also use the relation between molecular numbers and molecular concentrations as

$$[i] = X_i(t)/V, \quad i = S, E, C, P. \quad (7.4.8)$$

Applying (7.4.7) and (7.4.8) in (7.4.6), and cancelling out  $V$ , we get

$$X_{E_0} \ll X_{S_0} + \frac{\kappa'_{-1} + \kappa'_2}{\kappa'_1}. \quad (7.4.9)$$

Plugging our choice of the scaled variables and rate constants given in (7.3.2) and (7.4.3) to (7.4.9) gives

$$Z_E^N(t) + Z_C^N(t) \ll N \left( Z_S^N(t) + Z_P^N(t) \right) + Z_C^N(t) + \frac{N(\kappa_{-1} + \kappa_2)}{\kappa_1}. \quad (7.4.10)$$

Since  $Z_i^N(t) \approx O(1)$  and  $\kappa_k \approx O(1)$ , the left and the right sides of (7.4.10) become of order 1 and  $N$ , respectively. Therefore, our choice of scaling is in agreement with the conditions for the validity of the sQSSA in the deterministic model (7.4.6).

Note that the choice of scaling exponents in (7.4.3) is, in general, not unique. We now derive more general conditions on the scaling exponents,  $\alpha$ 's and  $\beta$ 's, leading to the sQSSA limit (7.4.5). Note that for (7.4.5) to hold the time scale of  $C$  should be faster than that of  $S$ , so that we can obtain (7.4.4) from the equation of  $C$ , i.e.,

$$\alpha_C - \max(\rho_1, \rho_{-1}, \rho_2) < \alpha_S - \max(\rho_1, \rho_{-1}), \quad (7.4.11)$$

which is an analogue of  $\frac{d}{dt}[C] \approx 0$ . Moreover, for  $E$  to be expressed in terms of  $C$  and retained in the limit, the species copy number of  $C$  has to be greater than or equal to that of  $E$  in the conservation equation of the total enzyme

$$\alpha_E \leq \alpha_C. \quad (7.4.12)$$

Finally, since all propensities are of the same order, all the terms are present in (7.4.5)

$$\rho_1 = \rho_{-1} = \rho_2. \quad (7.4.13)$$

Combining (7.4.11), (7.4.12), and (7.4.13) together, we get the following conditions

$$\alpha_E \leq \alpha_C < \alpha_S, \quad \alpha_S + \beta_1 = \beta_{-1} = \beta_2. \quad (7.4.14)$$

The second condition in (7.4.14) can be rewritten as  $\alpha_S = \beta_{-1} - \beta_1 = \beta_2 - \beta_1$  and so (7.4.14) implies

$$X_{E_0} \ll X_{S_0}, \quad X_{E_0} \ll \frac{\kappa'_{-1}}{\kappa'_1} \approx \frac{\kappa'_2}{\kappa'_1},$$

which is comparable to the general condition (7.4.6) on the deterministic sQSSA.

## 7.5 TOTAL QUASI-STEADY-STATE APPROXIMATION

In the deterministic tQSSA, we define the total substrate concentration as  $[T] := [S] + [C]$ . The idea behind the tQSSA is to get an accurate approximation for a wider range of the parameters (for example, covering both high and low enzyme concentrations). Assuming that  $[T]$  changes on the slow time scale, the equations (7.2.2)-(7.2.3) give the following reduced model (Borghans, De Boer, and L. A. Segel 1996; Tzafriri 2003),

$$\begin{aligned} \frac{d}{dt}[T] &= -k_2[C], \\ \frac{d}{dt}[C] &= k_1 \{([T] - [C])([E_0] - [C]) - K_M[C]\}, \end{aligned} \quad (7.5.1)$$

where  $K_M = (k_{-1} + k_2)/k_1$ . Assuming that  $\frac{d}{dt}[C] \approx 0$  and using  $[C] \leq [E_0]$ , the unique solution is found as the positive root of a quadratic equation

$$\frac{d}{dt}[C] = \frac{([E_0] + K_M + [T]) - \sqrt{([E_0] + K_M + [T])^2 - 4[E_0][T]}}{2}, \quad (7.5.2)$$

and the evolution of the total substrate concentration obeys

$$\frac{d}{dt}[T] = -k_2 \frac{([E_0] + K_M + [T]) - \sqrt{([E_0] + K_M + [T])^2 - 4[E_0][T]}}{2}. \quad (7.5.3)$$

The above approximation is the tQSSA of the MM kinetics (7.2.1) in the deterministic setting.

Now, consider the stochastic model (7.3.1). Our goal is to apply the multi-scale approximation with the appropriate scaling so that we can consider (7.5.3) as the limit of



the stochastic MM system (7.3.3) as  $N \rightarrow \infty$ . We assume that  $S$ ,  $E$ , and  $C$  are on the faster time scale than  $P$ . Our choice of scaling is

$$\alpha_S = \alpha_E = \alpha_C = \alpha_P = 1, \quad \beta_1 = \beta_2 = 0, \quad \beta_{-1} = 1, \quad (7.5.4)$$

that is,  $\rho_1 = \alpha_S + \alpha_E + \beta_1 = 2$ ,  $\rho_{-1} = \alpha_C + \beta_{-1} = 2$ , and  $\rho_2 = \alpha_C + \beta_2 = 1$ . We are interested in the stochastic model in the time scale of  $T$ . Adding unscaled equations for  $S$  and  $C$  and dividing by  $N^{\max(\alpha_S, \alpha_C)}$  from (7.3.3) we have

$$\begin{aligned} \frac{N^{\alpha_S} Z_S^{N,\gamma}(t) + N^{\alpha_C} Z_C^{N,\gamma}(t)}{N^{\max(\alpha_S, \alpha_C)}} &= \frac{N^{\alpha_S} Z_S^N(0) + N^{\alpha_C} Z_C^N(0)}{N^{\max(\alpha_S, \alpha_C)}} \\ &\quad - \frac{1}{N^{\max(\alpha_S, \alpha_C)}} Y_2 \left( \int_0^t N^{\rho_2 + \gamma} \kappa_2 Z_C^{N,\gamma}(s) ds \right). \end{aligned}$$

Thus, the time scale of  $T$  is given by

$$\gamma = \max(\alpha_S, \alpha_C) - \rho_2. \quad (7.5.5)$$

Using (7.5.4) gives  $\gamma = 0$ . For simplicity, we set the time scale exponent as  $\gamma = 0$  and denote  $Z_i^{N,\gamma}$  as  $Z_i^N$  for  $i = S, E, C, P$  as we did in Section 7.4.

Define the new slow variable

$$Z_T^N(t) := Z_S^N(t) + Z_C^N(t),$$

which satisfies

$$Z_T^N(t) = Z_T^N(0) - \frac{1}{N} Y_2 \left( \int_0^t N \kappa_2 Z_C^N(s) ds \right). \quad (7.5.6)$$

We have two conservation laws for the total amount of substrate and enzyme,  $m^N := Z_E^N(t) + Z_C^N(t)$  and  $k^N := Z_T^N(t) + Z_P^N(t)$ , and we denote their limits as  $N \rightarrow \infty$  by  $m$  and  $k$ , respectively. We also define

$$\mathbb{Z}_C^N(t) := \int_0^t Z_C^N(s) ds = m^N t - \int_0^t Z_E^N(s) ds.$$

Since  $Z_T^N(t) \leq k^N \rightarrow k$  and  $\mathbb{Z}_C^N(t) \leq m^N t \rightarrow mt$ ,  $Z_T^N$  and  $\mathbb{Z}_C^N$  are bounded, they are also relatively compact in the finite time interval  $t \in [0, T]$  where  $0 < T < \infty$ . Since the law of large numbers implies that  $Z_T^N(0) \rightarrow Z_T(0)$  as  $N \rightarrow \infty$  then  $(Z_T^N, \mathbb{Z}_C^N)$  (possibly along a subsequence only) converges to  $(Z_T, \mathbb{Z}_C)$  which satisfies

$$\begin{aligned} Z_T(t) &= Z_T(0) - \int_0^t \kappa_2 \dot{\mathbb{Z}}_C(s) ds, \\ 0 &= \int_0^t \kappa_1 (Z_T(s) - \dot{\mathbb{Z}}_C(s)) (m - \dot{\mathbb{Z}}_C(s)) ds - \int_0^t \kappa_{-1} \dot{\mathbb{Z}}_C(s) ds. \end{aligned} \quad (7.5.7)$$

Note that (7.5.7) is the limit as  $N \rightarrow \infty$  when we divide the equation for the scaled variable of  $C$  in (7.3.3) by  $N$ . Hence, we obtain

$$\dot{\mathbb{Z}}_C(t) = \frac{(m + \kappa_D + Z_T(t)) - \sqrt{(m + \kappa_D + Z_T(t))^2 - 4mZ_T(t)}}{2}, \quad (7.5.8)$$

$$\dot{Z}_T(t) = -\kappa_2 \frac{(m + \kappa_D + Z_T(t)) - \sqrt{(m + \kappa_D + Z_T(t))^2 - 4mZ_T(t)}}{2}, \quad (7.5.9)$$

where  $\kappa_D := \kappa_{-1}/\kappa_1$ . The equations (7.5.8) and (7.5.9) are analogous to (7.5.2) and (7.5.3), respectively. Note that we only have  $\kappa_D$  in (7.5.8)-(7.5.9) instead of  $K_M = (k_{-1} + k_2)/k_1$  in (7.5.2)-(7.5.3). The reaction rate  $\kappa_2$  disappears, since the propensity of the second reaction is of order of  $N$ , which is slower than the other two reactions whose propensities are of order  $N^2$  as shown in (7.3.3). In Figure 7.2, we compare the limit  $Z_T(t)$  in (7.5.9) and the scaled total substrate copy number  $Z_T^N(t)$  in (7.5.6), obtained from 1000 realisations of the stochastic simulation using Gillespie's algorithm (Gillespie 1977). The plot indicates close agreement between the scaled process  $Z_T^N(t)$  and its proposed limit  $Z_T(t)$ .

**CONDITIONS FOR tQSSA IN THE DETERMINISTIC SYSTEM** In order to derive tQSSA from (7.5.1), it is assumed that the total substrate concentration changes in the slow time scale and that the complex reaches its steady state quickly after some transient time, that is,  $\frac{d}{dt}[C] \approx 0$ . The complex concentration  $[C]$  is then found as the nonnegative solution of a quadratic equation. As mentioned earlier, the authors in Borghans, De Boer, and L. A. Segel (1996) approximated  $[C]$  in a form simpler than the exact solution in (7.5.2) and found a necessary and sufficient condition for the validity of the tQSSA as

$$K[E_0] \ll ([E_0] + [S_0] + K_M)^2, \quad (7.5.10)$$

where  $K = k_2/k_1$  and  $K_M = (k_{-1} + k_2)/k_1$ . The benefit of tQSSA over sQSSA is that (7.5.10) is always roughly valid (Pedersen, Bersani, and Bersani 2006; Tzafriri 2003). The condition (7.5.10) is equivalent to

$$1 \ll \left(1 + \frac{[E_0] + [S_0]}{K} + \frac{k_{-1}}{k_2}\right) \left(1 + \frac{[S_0] + K_M}{[E_0]}\right) \quad (7.5.11)$$

and is implied by any one of the following

$$K \ll [E_0] + [S_0], \quad k_2 \ll k_{-1}, \quad \text{and} \quad [E_0] \ll [S_0] + K_M. \quad (7.5.12)$$

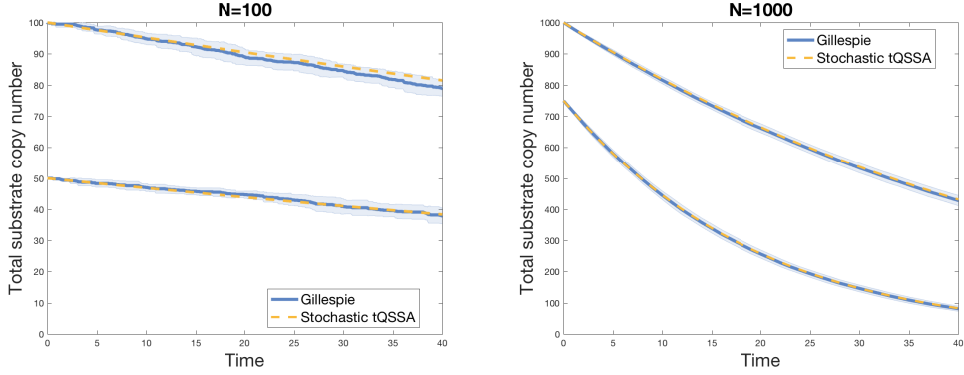
We convert concentrations and deterministic rate constants to molecular numbers and stochastic rate constants using (7.4.7)-(7.4.8). Simplifying, the condition in (7.5.10) becomes

$$\frac{\kappa'_2}{\kappa'_1} X_{E_0} \ll \left(X_{E_0} + X_{S_0} + \frac{\kappa'_{-1} + \kappa'_2}{\kappa'_1}\right)^2, \quad (7.5.13)$$

by using the same argument as in (7.4.9). Plugging our choice of the scaled variables and rate constants as specified in (7.3.2) and (7.5.4) yields

$$\begin{aligned} \frac{\kappa_2}{\kappa_1} N \left( Z_E^N(t) + Z_C^N(t) \right) &\ll \left( N \left( Z_E^N(t) + Z_C^N(t) \right) + N \left( Z_S^N(t) + Z_C^N(t) + Z_P^N(t) \right) \right. \\ &\quad \left. + \frac{N\kappa_{-1} + \kappa_2}{\kappa_1} \right)^2. \end{aligned} \quad (7.5.14)$$

Since in the above expression the term on the left is  $O(N)$  and the term on the right is  $O(N^2)$ , our choice of scaling in the stochastic model is in agreement with the condition (7.5.10) for the validity of the tQSSA in the deterministic model.



**Figure 7.2:** MM kinetics with tQSSA. The scaling limit of the total substrate copy number, drawn in yellow dotted line, is compared with the mean total substrate copy number, obtained from simulations using the Gillespie's algorithm and shown in blue. The light blue shaded region represents one standard deviation from the mean. Different choices of initial conditions are used to reflect the fact that convergence can be achieved under varying values of the conservation constant  $m$ . Simulation settings: **(a)**  $N = 100$ ,  $(X_S^N(0), X_E^N(0), X_C^N(0), X_P^N(0)) = (100, 10, 0, 0)$  for the upper curve and  $(50, 10, 0, 0)$  for the lower curve; and **(b)**  $N = 1000$ ,  $(X_S^N(0), X_E^N(0), X_C^N(0), X_P^N(0)) = (1000, 10, 0, 0)$  for the upper curve and  $(750, 25, 0, 0)$  for the lower curve. The reaction rate constants are  $(\kappa_1, \kappa_{-1}, \kappa_2) = (1, 4, 1)$  in both **(a)** and **(b)**.

We may also derive more general conditions on the scaling exponents,  $\alpha$ 's and  $\beta$ 's, which lead to tQSSA limit in (7.5.9). To this end note that the time scale of  $C$  is faster than that of  $T$  so that we can derive an analogue of  $\frac{d}{dt}[C] \approx 0$  in (7.5.7)

$$\alpha_C - \max(\rho_1, \rho_{-1}, \rho_2) < \max(\alpha_S, \alpha_C) - \rho_2. \quad (7.5.15)$$

Moreover, the species copy number of  $C$  has an order greater than or equal to that of  $S$ , since otherwise  $C$  would disappear in the limit of  $T$ . Similarly, the species copy number of  $C$  has an order greater than or equal to that of  $E$  so that the limit for  $E$  can be expressed in terms of a conservation constant and  $C$ . Therefore, we have

$$\max(\alpha_S, \alpha_E) \leq \alpha_C. \quad (7.5.16)$$

Finally, to obtain a quadratic equation with a square root solution in the limit, the enzyme binding reaction rate should be equal to the unbinding reaction rate. That is,

$$\rho_1 = \rho_{-1}. \quad (7.5.17)$$

Combining (7.5.15), (7.5.16), and (7.5.17), we get the following conditions

$$\max(\alpha_S, \alpha_E) \leq \alpha_C, \text{ and } \beta_2 < \beta_{-1} = \alpha_C + \beta_1. \quad (7.5.18)$$

Note that due to  $\beta_2 < \beta_{-1}$  in (7.5.18), we have the discrepancy between  $\kappa_D$  in (7.5.9) and  $K_M$  in (7.5.3). In other words, the reason behind this discrepancy is that the propensity of the second reaction (product formation) is of order of  $N$ , which is slower than the

other two reactions whose propensities are of order  $N^2$  as shown in (7.3.3). Therefore, the reaction rate  $\kappa_2$  disappears. The condition (7.5.18) implies

$$X_{S_0} \approx X_{E_0}, \text{ and } \frac{\kappa'_2}{\kappa'_1} \ll \frac{\kappa'_{-1}}{\kappa'_1} \approx X_{E_0}, \quad (7.5.19)$$

which is consistent with the condition  $k_2 \ll k_{-1}$  in (7.5.12) that was also suggested for the stochastic system tQSSA in Barik et al. (2008).

## 7.6 REVERSE QUASI-STEADY-STATE APPROXIMATION

In the deterministic rQSSA, it is assumed that the enzyme is in high concentration. In this approximation, two time scales are considered. Starting with an initial condition  $([S], [E], [C], [P]) = ([S_0], [E_0], 0, 0)$  in (7.2.2), the enzyme concentration is  $[E] \approx [E_0]$  during the initial transient phase. Since there is almost no complex during this time, we get an approximate model as

$$\frac{d}{dt}[S] = -k_1[E_0][S], \quad \frac{d}{dt}[C] = k_1[E_0][S]. \quad (7.6.1)$$

After the initial transient phase, the substrate is depleted. Therefore, we assume that  $\frac{d}{dt}[S] \approx 0$  in (7.2.2) and obtain

$$[S] = \frac{k_{-1}[C]}{k_1([E_0] - [C])}, \quad (7.6.2)$$

so that the differential equation for the complex becomes

$$\frac{d}{dt}[C] = -k_2[C]. \quad (7.6.3)$$

We refer to the approximation of the system (7.2.2) by (7.6.1)-(7.6.3) as the rQSSA of the MM kinetics in the deterministic setting.

As in the previous sections, let us consider the stochastic equations for the MM kinetics given by (7.3.1) and again apply yet another multi-scale approximation with time change, to derive the rQSSA in (7.6.1)-(7.6.3). Assuming that  $S$  and  $C$  are on faster time scale than  $E$  and  $P$ , the following scales are chosen

$$\alpha_S = \alpha_C = \alpha_P = 1, \quad \alpha_E = 2, \quad \beta_1 = 0, \quad \beta_{-1} = \beta_2 = 1, \quad (7.6.4)$$

that is,  $\rho_1 = \alpha_S + \alpha_E + \beta_1 = 3$ ,  $\rho_{-1} = \alpha_C + \beta_{-1} = 2$ , and  $\rho_2 = \alpha_C + \beta_2 = 2$ . Note that this choice of scaling does not satisfy the balance equations introduced in (7.3.7). The inequalities for  $S$  and  $C$  give  $\gamma \leq -2$  and those for  $E$  and  $P$  give  $\gamma \leq -1$ . These conditions suggest the first and the second time scales as  $\gamma = -2$  when  $S$  and  $C$  become  $O(1)$  and  $\gamma = -1$  when  $E$  and  $P$  are  $O(1)$ . Define the following conservation constants

$$m^N = Z_E^{N,\gamma}(t) + \frac{1}{N}Z_C^{N,\gamma}(t), \quad k^N = Z_S^{N,\gamma}(t) + Z_C^{N,\gamma}(t) + Z_P^{N,\gamma}(t), \quad (7.6.5)$$

which we assume to converge to some limiting values  $m$  and  $k$  as  $N \rightarrow \infty$ , respectively. In this setting,  $Z_S^{N,\gamma}$ ,  $Z_E^{N,\gamma}$ ,  $Z_C^{N,\gamma}$ , and  $Z_P^{N,\gamma}$  are bounded so that they are relatively compact for  $t \in [0, \mathcal{T}]$ , where  $0 < \mathcal{T} < \infty$ . In the first time scale when  $\gamma = -2$ , the

scaled species for  $E$  and  $P$  converge to their initial conditions,  $Z_E^{N,-2}(t) \rightarrow Z_E(0)$  and  $Z_P^{N,-2}(t) \rightarrow Z_P(0)$  as  $N \rightarrow \infty$ , since the scaling exponents in the propensities are greater than those of species copy numbers in this time scale. Therefore  $(Z_S^{N,-2}, Z_C^{N,-2})$  converges to  $(Z_S^{(-2)}, Z_C^{(-2)})$  satisfying

$$\begin{aligned} Z_S^{(-2)}(t) &= Z_S(0) - \int_0^t \kappa_1 Z_S^{(-2)}(s) Z_E(0) ds, \\ Z_C^{(-2)}(t) &= Z_C(0) + \int_0^t \kappa_1 Z_S^{(-2)}(s) Z_E(0) ds. \end{aligned} \quad (7.6.6)$$

Since  $Z_C^{N,-2}(t)$  is bounded by  $k^N$  from (7.6.5), the remaining reaction terms for the unbinding of the complex and for the product production vanish as  $N \rightarrow \infty$ . The equations (7.6.6) are seen as the integral version of (7.6.1), that is, the rQSSA for the first (transient) time scale.

Next, consider the second time scale when  $\gamma = -1$ . Plugging  $\gamma = -1$  in the equation for  $S$  in (7.3.3), and applying the law of large numbers, we obtain

$$Z_S^{N,-1}(t) \approx Z_S^N(0) - \int_0^t \left( N \kappa_1 Z_S^{N,-1}(s) Z_E^{N,-1}(s) - \kappa_{-1} Z_C^{N,-1}(s) \right) ds. \quad (7.6.7)$$

Using (7.6.7), the equations for  $E$  and  $C$  in (7.3.3) become

$$Z_C^{N,-1}(t) \approx Z_C^N(0) + Z_S^N(0) - Z_S^{N,-1}(t) - \int_0^t \kappa_2 Z_C^{N,-1}(s) ds, \quad (7.6.8)$$

$$Z_E^{N,-1}(t) \approx Z_E^N(0) - \int_0^t \kappa_1 Z_S^{N,-1}(s) Z_E^{N,-1}(s) ds, \quad (7.6.9)$$

since the remaining reaction terms are asymptotically equal to zero. Dividing (7.6.7) by  $N$ , we obtain

$$\int_0^t \kappa_1 Z_S^{N,-1}(s) Z_E^{N,-1}(s) ds \rightarrow 0, \quad (7.6.10)$$

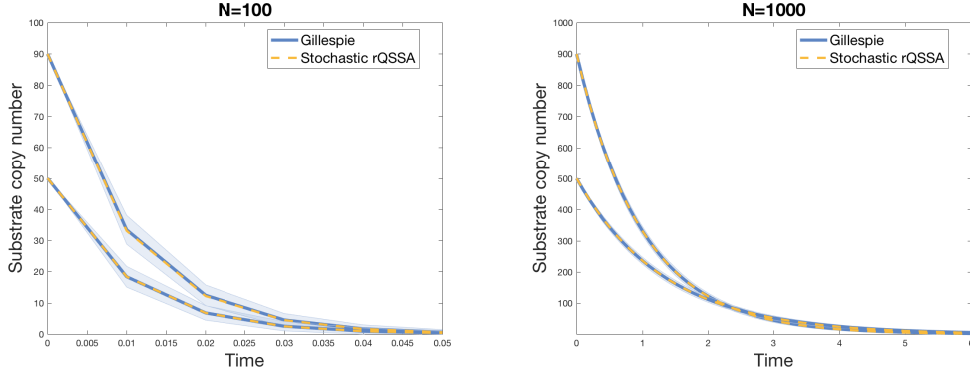
as  $N \rightarrow \infty$ , since all other terms vanish asymptotically. Due to (7.6.9) and (7.6.10),  $Z_E^{N,-1}(t) \rightarrow Z_E(0)$  as  $N \rightarrow \infty$ . Defining  $\mathbb{Z}_S^{N,-1}(t) := \int_0^t Z_S^{N,-1}(s) ds$  and using (7.6.10) and (7.6.8), we conclude that  $(\mathbb{Z}_S^{N,-1}, Z_C^{N,-1})$  converges to  $(\mathbb{Z}_S^{(-1)}, Z_C^{(-1)})$  satisfying

$$\begin{aligned} 0 &= \int_0^t \kappa_1 \dot{\mathbb{Z}}_S^{(-1)}(s) Z_E(0) ds, \\ Z_C^{(-1)}(t) &= Z_C(0) + Z_S(0) - \mathbb{Z}_S^{(-1)}(t) - \int_0^t \kappa_2 Z_C^{(-1)}(s) ds. \end{aligned} \quad (7.6.11)$$

Therefore,

$$\dot{\mathbb{Z}}_S^{(-1)}(t) = 0, \quad \dot{Z}_C^{(-1)}(t) = -\kappa_2 Z_C^{(-1)}(t), \quad (7.6.12)$$

which is the analogue of the rQSSA in the second time scale (7.6.2)-(7.6.3) as derived from the deterministic model.



**Figure 7.3:** MM kinetics with rQSSA in the first time scale  $\gamma = -2$ : The scaling limit of the substrate copy number, drawn in yellow dotted line, is compared with the mean substrate copy number, obtained from simulations using the Gillespie's algorithm and shown in blue. The light blue shaded region represents one standard deviation from the mean. Simulation settings: **(a)**  $N = 100$ ,  $(X_S^N(0), X_E^N(0), X_C^N(0), X_P^N(0)) = (90, 10^6, 10, 0)$  for the upper curve and  $(50, 10^6, 10, 0)$  for the lower curve; and **(b)**  $N = 1000$ ,  $(X_S^N(0), X_E^N(0), X_C^N(0), X_P^N(0)) = (900, 10^6, 10, 0)$  for the upper curve and  $(500, 75 \cdot 10^4, 110, 0)$  for the lower curve. The reaction rate constants are  $(\kappa_1, \kappa_{-1}, \kappa_2) = (1, 1, 0.1)$  in both **(a)** and **(b)**. Given the scaling assumptions, the convergence is not sensitive to the exact values of the initial conditions. The only purpose of the two different sets of initial conditions is to illustrate convergence under varying values of the conservation constant  $m$ .

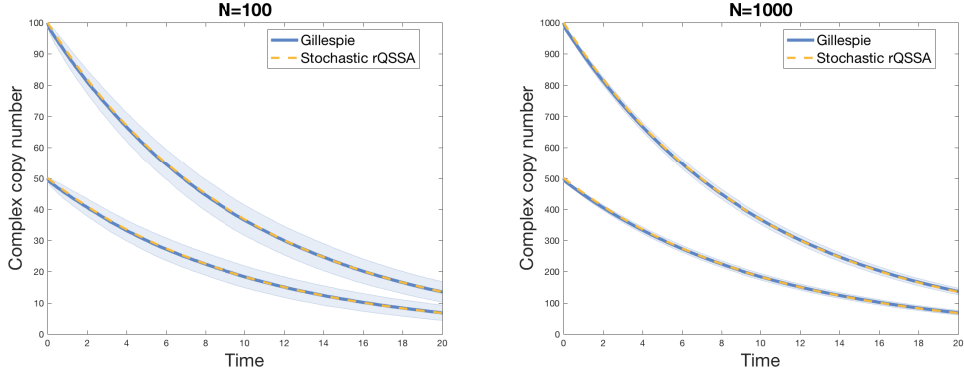
We illustrate the quality of rQSSA in the stochastic MM system with some simulations. In Figure 7.3, we compare the limit  $Z_S^{(-2)}(t)$  in (7.6.6) and the scaled substrate copy number  $Z_S^{N,-2}(t)$  in (7.3.3) using 1000 runs of the Gillespie's algorithm. In Figure 7.4, we compare the limit  $Z_C^{(-1)}(t)$  in (7.6.12) and the scaled complex copy number  $Z_C^{N,-1}(t)$  in (7.3.3) using 10000 runs of the Gillespie's algorithm. Note that the initial condition of  $Z_C^{(-1)}(t)$  is  $Z_C(0) + Z_S(0)$  in (7.6.11). However, this does not affect since  $Z_S(0) = 0$  in our simulation in Figure 7.4. In both time scales, the scaled processes are in close agreement with the proposed limits.

**CONDITIONS FOR RQSSA IN THE DETERMINISTIC SYSTEM.** Consider the general condition for the validity of the rQSSA at high enzyme concentrations suggested by Schnell and Maini (2000),

$$K \ll [E_0] \quad \text{and} \quad [S_0] \ll [E_0], \quad (7.6.13)$$

where  $K = k_2/k_1$ . Rewriting (7.6.13) in terms of molecular copy numbers and stochastic rate constants using (7.4.7)-(7.4.8) gives

$$\frac{\kappa'_2}{\kappa'_1} \ll X_{E_0} \quad \text{and} \quad X_{S_0} \ll X_{E_0}, \quad (7.6.14)$$



**Figure 7.4:** MM kinetics with rQSSA in the second time scale  $\gamma = -1$ : The scaling limit of the complex copy number, drawn in yellow dotted line, is compared with the mean complex copy number, obtained from simulations using the Gillespie's algorithm and shown in blue. The light blue shaded region represents one standard deviation from the mean. Simulation settings: **(a)**  $N = 100$ ,  $(X_S^N(0), X_E^N(0), X_C^N(0), X_P^N(0)) = (0, 10^4, 100, 0)$  for the upper curve and  $(0, 7500, 50, 0)$  for the lower curve; and **(b)**  $N = 1000$ ,  $(X_S^N(0), X_E^N(0), X_C^N(0), X_P^N(0)) = (0, 10^5, 10^3, 0)$  for the upper curve and  $(0, 75 \cdot 10^3, 500, 0)$  for the lower curve. The reaction rate constants are  $(\kappa_1, \kappa_{-1}, \kappa_2) = (1, 1, 0.1)$  in both **(a)** and **(b)**. Given the scaling assumptions, the convergence is not sensitive to the exact values of the initial conditions. The only purpose of the two different sets of initial conditions is to illustrate convergence under varying values of the conservation constant  $m$ .

since  $V$ 's all cancel out. Using our choice of scaling in (7.3.2) and (7.6.4), the conditions (7.6.14) become

$$\begin{aligned} \frac{N\kappa_2}{\kappa_1} &\ll \left( N^2 Z_E^{N,\gamma}(t) + N Z_C^{N,\gamma}(t) \right) \quad \text{and} \\ N \left( Z_S^{N,\gamma}(t) + Z_C^{N,\gamma}(t) + Z_P^{N,\gamma}(t) \right) &\ll \left( N^2 Z_E^{N,\gamma}(t) + N Z_C^{N,\gamma}(t) \right). \end{aligned} \quad (7.6.15)$$

Since the inequalities in (7.6.15) hold for large  $N$ , our choice of scaling is seen to satisfy the conditions (7.6.13).

As seen in the previous sections, we may also derive more general conditions on the scaling exponents,  $\alpha$ 's and  $\beta$ 's, leading to (7.6.6) and (7.6.12). In the first scaling, the time scales of  $S$  and  $C$  are the same and faster than the time scale of  $E$ . Therefore it follows

$$\alpha_S - \max(\rho_1, \rho_{-1}) = \alpha_C - \max(\rho_1, \rho_{-1}, \rho_2) < \alpha_E - \max(\rho_1, \rho_{-1}, \rho_2). \quad (7.6.16)$$

Since the binding reaction rate of the enzyme is faster than the rates of the other two reactions as we see in the limit (7.6.6), we have

$$\max(\rho_{-1}, \rho_2) < \rho_1. \quad (7.6.17)$$

Combining (7.6.16) and (7.6.17), the conditions in the first time scale are

$$\alpha_S = \alpha_C < \alpha_E, \quad \max(\beta_{-1}, \beta_2) < \alpha_E + \beta_1. \quad (7.6.18)$$

Then, the condition in (7.6.18) implies

$$X_{S_0} \ll X_{E_0}, \quad \max \left( \frac{\kappa'_{-1}}{\kappa'_1}, \frac{\kappa'_2}{\kappa'_1} \right) \ll X_{E_0}, \quad (7.6.19)$$

which is comparable to (7.6.13). Next, consider the second time scale and the condition on the scaling exponents that yields (7.6.12). Note that the conditions (7.6.16)-(7.6.17) are already sufficient to derive the limiting process in the second time scale. The condition (7.6.16) implies the time scales of  $S$  and  $C$  are the same. Since  $\rho_2 < \rho_1$  as in (7.6.17), the time scale of  $S + C$  is slower than that of  $S$ . Setting the time scale of  $S + C$  as the reference one, we see that on that timescale  $S$  will be rapidly depleted and then approximated by zero in view of the discrepancy between the consumption and production rates of  $S$ , due to  $\rho_{-1} < \rho_1$  in (7.6.17). Therefore, the conditions in (7.6.16)-(7.6.17) are sufficient to obtain the limit in (7.6.12) on the second time scale as well. Finally, note that the stochastic MM system with (7.6.18) does not provide an analogue equation for  $S$  in (7.6.2) due to the condition,  $\rho_{-1} < \rho_1$ , as shown in (7.6.17). Assuming  $\rho_{-1} = \rho_1$  will balance the production and the consumption of  $S$ , but in this case we can no longer claim the relative compactness of  $S$ .

## 7.7 DISCUSSION

Our derivations in this chapter rely on the multi-scale approximation approach (Ball et al. 2006; Kang and Kurtz 2013) that is quite general and could be used to obtain similar types of QSSAs in other more general stochastic CRNs. As an illustration, we have briefly considered the ESI system (L. A. Segel 1988) and derived various QSSAs in Appendix E.1. Other QSSAs (also for other variants of the ESI system) can be derived similarly. Another important application area where our tools can be used to derive meaningful approximations is a model of signal transduction into protein phosphorylation cascade, such as the Mitogen-activated Protein Kinase (MAPK) signalling pathway (Bersani, Pedersen, et al. 2005; Dell'Acqua and Bersani 2011; Gómez-Urbe, Verghese, and Mirny 2007). In MAPK signalling pathway, the product of one level of the cascade may act as the enzyme at the next level, with different MM QSSAs found to be appropriate at different levels (Bersani, Pedersen, et al. 2005; Dell'Acqua and Bersani 2011; Gómez-Urbe, Verghese, and Mirny 2007; Sauro and Kholodenko 2004). Our tools can provide further insights into the biophysics of such systems.

Since the dynamics of enzyme kinetics plays such a central role in many problems of modern biochemistry, it is important to understand the precise conditions for the QSSAs discussed here. For convenience, in Table 7.2, we summarise the conditions for different QSSAs in terms of their scaling exponents as well as the stochastic and deterministic species abundances. The conditions for the stochastic scalings presented in the first row of the table clearly separate the range of parameter values into three regimes. As we can see, the exponent  $\alpha_S$  should be greater than the other exponents for species copy numbers in the sQSSA while  $\alpha_E$  is greater than the other exponents for species copy numbers in the rQSSA. In the tQSSA,  $\alpha_C$  needs to be greater than or equal to the other exponents. For the sQSSA and the rQSSA, the stochastic species abundance conditions (listed in the second row) are seen to also imply the deterministic abundance conditions (listed in the third row). However, the necessary condition for the



**Table 7.2:** Comparison of conditions for the quasi-steady-state approximations in the stochastic and deterministic MM kinetics.

Conditions	sQSSA	tQSSA	rQSSA
stochastic scaling	$\alpha_E \leq \alpha_C < \alpha_S$ $\alpha_S = \beta_{-1} - \beta_1 = \beta_2 - \beta_1$	$\max(\alpha_S, \alpha_E) \leq \alpha_C$ $\beta_2 < \beta_{-1} = \alpha_C + \beta_1$	$\alpha_S = \alpha_C < \alpha_E$ $\max(\beta_{-1}, \beta_2) < \alpha_E + \beta_1$
stochastic abundance	$X_{E_0} \ll X_{S_0}$ $X_{E_0} \ll \frac{\kappa'_{-1}}{\kappa'_1} \approx \frac{\kappa'_2}{\kappa'_1}$	$X_{E_0} \approx X_{S_0}$ $\frac{\kappa'_2}{\kappa'_1} \ll \frac{\kappa'_{-1}}{\kappa'_1} \approx X_{E_0}$	$X_{S_0} \ll X_{E_0}$ $\max\left(\frac{\kappa'_{-1}}{\kappa'_1}, \frac{\kappa'_2}{\kappa'_1}\right) \ll X_{E_0}$
deterministic abundance	$[E_0] \ll [S_0] + K_M$	$K[E_0] \ll ([E_0] + [S_0] + K_M)^2$	$K \ll [E_0]$ and $[S_0] \ll [E_0]$

The parameters are  $K = k_2/k_1$  and  $K_M = (k_{-1} + k_2)/k_1$ .

tQSSA derived from the stochastic model is slightly different from the corresponding deterministic condition as it requires similar order of magnitude for the total amount of enzyme and the total amount of substrate. Note, however, that the condition on the deterministic rates  $k_2 \ll k_{-1}$ , which is an analogue of the stochastic rates condition  $\kappa'_2 \ll \kappa'_{-1}$ , implies both the deterministic and the stochastic abundance conditions for the tQSSA.

Our derivations of the QSSAs from the stochastic MM kinetics provide approximate ODE models where reaction propensities follow rational or square-root functions and hence violate the law of mass action. Such non-standard propensity functions are often useful for building efficient reduced model also in the stochastic settings where they may be used as intensity functions in the random time change representation of the Poisson processes. For instance, Grima, Schmidt, and Newman (2012), Choi, Rempała, and J. Kim (2017), as well as some others H. Kim and Gelenbe (2012) and Tian and Burrage (2006) have applied this idea to construct approximate, stochastic MM enzyme kinetic networks and even the gene regulatory networks (Smith, Cianci, and Grima 2016). As some of the authors of this article argued in their recent work (see J. K. Kim, Rempała, and Kang (2017)), such approximate stochastic models using intensities derived from the deterministic limits may in some sense be better approximations of the underlying stochastic networks than the deterministic QSSAs. Our derivations presented here could be used to further justify this statement, at least for networks satisfying certain scaling conditions (J. K. Kim, Josić, and Bennett 2015; Rao and Arkin 2003; Sanft, Gillespie, and Petzold 2011), including those presented in Table 7.2. We therefore hope that the results in the current work will further contribute to developing more accurate approximations of models for enzyme kinetics in biochemical networks.

With these QSSAs of the MM enzyme-kinetic CRNs, which, as a queueing system, models a special case of the collaborative uploading problem described in Section 1.1, we conclude our study of parallel queueing systems. In the next chapter, we shall begin our study of the MABMs. In particular, we shall derive an FCLT for the simplest MABM, an ID or an SI process on Configuration Model (CM) random graphs.

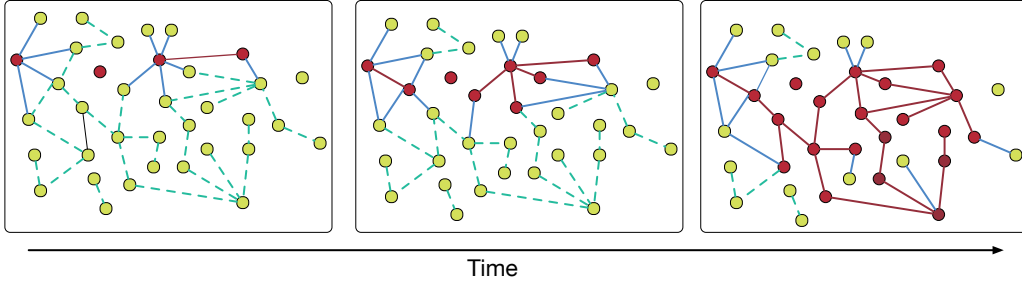
In this chapter, we begin our study of the distribution problem described in Section 1.1. The main modelling framework for the distribution problem is that of an MABM. Here, we analyse the simplest possible distribution process, namely the ID process, which keeps track of whether a piece of information has reached its intended destination or not. The intended destinations are considered to be vertices of a graph. From a mathematical perspective, this binary dynamical process is equivalent to the stochastic SI process from epidemiology literature. In this chapter, we shall derive a scaling limit for such a dynamical process. To be precise, we study a stochastic compartmental SI epidemic process on a CM random graph with a given degree distribution over a finite time interval  $\mathcal{T} := [0, T]$ , for some  $T > 0$  (see Figure 8.1). In this setting, we split the population into two compartments, namely,  $S$  and  $I$ , denoting the susceptible and infected individuals, respectively. In addition to the sizes of these two compartments, we consider counts of  $SI$ -edges (those connecting a susceptible and an infected individual) and  $SS$ -edges (those connecting two susceptible individuals). We describe the dynamical process in terms of these counts and present an FCLT for them as  $n$  grows to infinity. To be precise, we show that these counts, when appropriately scaled, converge weakly to a continuous Gaussian vector semimartingale process in the space of real 3-dimensional vector-valued càdlàg functions on  $\mathcal{T}$  endowed with the Skorohod topology.

## 8.1 MODEL

**ADDITIONAL CONVENTIONS** We denote by  $D = D(\mathcal{T})$  the Polish space of real functions  $f$  on  $\mathcal{T}$  that are right continuous and have left hand limits. A function  $f \in D$  is called càdlàg. Unless otherwise mentioned, the space  $D$  is assumed endowed with the Skorohod topology (Billingsley 1999, Chapter 3). Accordingly we call  $D$  the Skorohod space. Let the triplet  $(\Omega, \mathcal{F}, \mathbb{P})$  denote our probability space. For a differentiable function  $f$  defined on some  $E \subseteq \mathbb{R}^d$ , we denote its partial derivative with respect to the  $i$ -th variable by  $\partial_i f$ , for  $i = 1, 2, \dots, d$ . With some abuse of notation, we use  $\partial f(x)$  to denote the derivative of a differentiable function of a single variable at  $x$ . For a sequence of random variables  $\{Z_n\}_{n \in \mathbb{N}}$ , the phrase “ $Z_n \rightarrow \infty$  with high probability (whp)” means “ $\mathbb{P}(Z_n > k) \rightarrow 1$  as  $n \rightarrow \infty$  for any  $k > 0$ ”. For a stochastic process  $Z$  with paths in  $D$ , we denote its jump sizes by  $\delta Z$ .

We begin with the class of all CM random graphs (Hofstad 2017, Chapter 7) with  $n$  nodes, for  $n \in \mathbb{N}$ . The main advantage of the CM is that it allows one to fix the degrees before constructing the graph itself. There are numerous real life situations where random graphs with a prescribed degree sequence (or a distribution) are reasonable and intuitive. See Hofstad (2017, Chapter 7) for some examples.

Given a sequence of degrees  $(d_1, d_2, \dots, d_n)$  for  $n$  vertices, we first assign  $d_i$  half-edges to node  $i$ . The CM random graph is then obtained by uniformly matching (or pairing) all available half-edges. Two paired half-edges form an edge. To be precise, we actually



**Figure 8.1:** Dynamics of the stochastic SI model over a finite time interval  $\mathcal{T}$ . Susceptible nodes are shown in yellowish green and infected nodes, in dark red. Infected nodes infect their neighbours at a given rate  $\beta > 0$ . Different types of edges, namely, SI, SS, and II, are shown in different colours. In the context of non-equilibrium percolation, which we model as a stochastic SI process, the red nodes are the wet (or active) nodes and the growth of the red-component describes the time evolution of the percolated component.

get a *multigraph* because the resultant graph may have self-loops and multiple edges. However, we can circumvent this problem by conditioning on simplicity of the graph as  $n \rightarrow \infty$ . A graph is called simple if there is no self-loop and if there is at most one edge between any pair of vertices *i.e.*, multiple edges are not allowed. As shown in Janson (2009), the conditional probabilities can be calculated because the probability that the random multigraph is simple is strictly positive provided  $\sum_i d_i^2 = O(\sum_i d_i)$  (or equivalently, after ignoring isolated vertices,  $\sum_i d_i^2 = O(n)$ ). We assume this condition to be satisfied in our setting.

Let us denote the Probability Generating Function (PGF) of the underlying degree distribution by  $\psi$ , *i.e.*,

$$\psi(x) := \sum_k x^k p_k, \quad (8.1.1)$$

where  $p_k$  is the probability that a randomly chosen node has degree  $k$ . We denote the class of all CM random graphs with  $n$  nodes by  $\mathcal{G}(\psi, n)$ .

We consider the stochastic SI model on CM random graphs. Each infected individual (represented by a node of the graph) infects one of its neighbours at rate  $\beta > 0$ . We split the population into two compartments, namely,  $S$  and  $I$ , consisting respectively of the susceptible and infected individuals<sup>1</sup>.

Let  $X(t) := (X_S(t), X_{SI}(t), X_{SS}(t))$  denote the aggregated state vector of the system at time  $t \geq 0$ , where  $X_S(t)$ ,  $X_{SI}(t)$  and  $X_{SS}(t)$  respectively indicate the number of susceptible individuals, the number of SI edges (edges connecting an  $S$ -type individual and an  $I$ -type individual) and the number of SS edges. Please note that  $X_{SS}$  counts these edges twice. In order to describe the time evolution of these counts, we also need certain auxiliary counts. Denote by  $X_{SI,i}(t)$  and  $X_{SS,i}(t)$ , the numbers of infected and susceptible neighbours of a susceptible node  $i$  at time  $t$ , respectively. We shall often omit the time

<sup>1</sup> In the context of the distribution problem described in Section 1.1, the infected individuals are the ones who already received the content, and the susceptible vertices are yet to receive.

argument  $t$  if there is no ambiguity. Define the filtration  $\mathcal{F}_t$  as the  $\sigma$ -field generated by the process history up to and including time  $t > 0$  (Durrett 2010a, Chapter 1, p. 14). We let  $\mathcal{F}_0$  contain all  $\mathbb{P}$ -null sets in  $\mathcal{F}$ . We include all  $\mathbb{P}$ -null sets in  $\mathcal{F}$  so that the filtration family  $\{\mathcal{F}_t\}$  is complete. Also it is right continuous (i.e.,  $\mathcal{F}_{t+} = \mathcal{F}_t$  for every  $t \geq 0$ , where  $\mathcal{F}_{t+} := \bigcap_{s>0} \mathcal{F}_{t+s}$  is the  $\sigma$ -field of events immediately after  $t$ ), because it is generated by a right continuous jump process (Andersen et al. 1993, Chapter II, p. 61). Therefore, the usual Dellacherie's conditions on  $\{\mathcal{F}_t\}$  are satisfied (Fleming and Harrington n.d.; Karatzas and Shreve 1991; Rebolledo 1980). By the Doob-Meyer decomposition theorem (Meyer 1962), we decompose the semimartingale  $X$  as

$$X(t) = X(0) + \int_0^t \mathbb{F}_X(X(s)) ds + M'(t), \quad (8.1.2)$$

where  $M'(t) := (M'_S(t), M'_{SI}(t), M'_{SS}(t))$  is a zero-mean martingale adapted to the filtration  $\mathcal{F}_t$  and  $\mathbb{F}_X(X) := (\mathbb{F}_S(X_{SI}), \mathbb{F}_{SI}(X_{SS}, X_{SI}), \mathbb{F}_{SS}(X_{SS}, X_{SI}))$  is an integrable function given by

$$\begin{aligned} \mathbb{F}_S(X_{SI}) &:= -\beta X_{SI}, \\ \mathbb{F}_{SI}(X_{SS}, X_{SI}) &:= \sum_{i \in S} \beta X_{SI,i} (X_{SS,i} - X_{SI,i}), \\ \mathbb{F}_{SS}(X_{SS}, X_{SI}) &:= -2 \sum_{i \in S} \beta X_{SI,i} X_{SS,i}. \end{aligned} \quad (8.1.3)$$

Let  $X_{S\bullet}(t)$  be the number of edges between a susceptible node and a node of any other status at time  $t \geq 0$ . We partition the collection of susceptible nodes  $S$  by their degree  $k \in \mathbb{N}_0$  so that  $S = \bigcup_k S_k$ , where  $S_k$  is the collection of susceptible nodes of degree  $k$ . Therefore we have  $X_S = \sum_k X_{S_k}$ , where  $X_{S_k}$  is the size of  $S_k$ , and  $X_{S\bullet}(t) := \sum_k k X_{S_k}(t)$ . In order to study the large graph limit of the system, we also define the following quantity

$$\theta(t) := \exp\left(-\beta \int_0^t \frac{X_{SI}(s)}{X_{S\bullet}(s)} ds\right), \quad (8.1.4)$$

which can be intuitively described as the probability that a degree-1 node that was susceptible at time zero remains susceptible till time  $t > 0$  (Miller 2011; Miller, Slim, and E. M. Volz 2012; E. Volz 2008). It may be described equivalently as a solution to the following integral equation

$$\theta(t) = \theta(0) + \int_0^t \mathbb{F}_\theta(X_{SI}(s), X_{S\bullet}(s), \theta(s)) ds,$$

where  $\theta(0) = 1$  and  $\mathbb{F}_\theta(X_{SI}, X_{S\bullet}, \theta) := -\beta \theta \frac{X_{SI}}{X_{S\bullet}}$ .

## 8.2 THE LAW OF LARGE NUMBERS

We adopt the framework of Jacobsen et al. (2016) for our purpose and make the following technical assumptions. Unless otherwise stated, all limits below and elsewhere in this chapter are taken in the large graph limit, i.e., as  $n \rightarrow \infty$ . Let  $\mathcal{T}_0 := (0, T] \subset \mathcal{T}$ .

**A1** For  $t \in \mathcal{T}_0$ ,  $X_{S\bullet}(t) \rightarrow \infty$  whp. This assumption ensures the infection does not take over the entire graph and there are sufficiently many susceptible individuals throughout  $\mathcal{T}_0$ . Furthermore, the quantity  $\theta(t)$  remains well-defined on the entirety of  $\mathcal{T}$ .

**A2** The fraction of initially susceptible nodes converges to some  $\alpha_S$ , *i.e.*,

$$n^{-1}X_S(0) \xrightarrow{P} \alpha_S. \quad (8.2.1)$$

We also assume that the initially infected and susceptible nodes are selected uniformly at random and  $\alpha_S > 0$ . Note that, by virtue of uniformly random selection of infected nodes at time 0, the above also implies (see Jacobsen et al. (2016))

$$\begin{aligned} n^{-1}X_I(0) &\xrightarrow{P} \alpha_I = 1 - \alpha_S, \\ n^{-1}X_{SI}(0) &\xrightarrow{P} \alpha_{SI} = \alpha_S(1 - \alpha_S)\partial\psi(1), \\ n^{-1}X_{SS}(0) &\xrightarrow{P} \alpha_{SS} = \alpha_S^2\partial\psi(1). \end{aligned} \quad (8.2.2)$$

We shall use the vector notation  $\alpha = (\alpha_S, \alpha_{SI}, \alpha_{SS})$ . The process  $X_I$  captures the number of infected individuals.

**A3**  $\sum_k k^3 p_k < \infty$ .

Having laid down our technical assumptions, define the operator  $\mathbb{D}^r$  as

$$\mathbb{D}^r f := f^{r-1} \frac{\partial^r f}{(\partial f)^r}, \quad (8.2.3)$$

for  $f : \mathbb{R} \rightarrow \mathbb{R}$  and  $r \in \mathbb{N}$ , whenever the division of  $\partial^r f$  by  $(\partial f)^r$  makes sense, where  $f^r$  is understood as  $(r-1)$ -times multiplication of  $f$  with itself for  $r \in \mathbb{N}_0$  with the convention  $f^0 := 1$ . The symbol  $\partial^r f$  denotes the  $r$ -th derivative of the function  $f$ , and by convention, we write  $\partial f := \partial^1 f$ . The operators  $\mathbb{D}^r$  are used to capture the impact of the graph structure on the limiting dynamics through the degree distribution. Then, define  $(x, \vartheta) := ((x_S, x_{SI}, x_{SS}), \vartheta)$ , and  $\kappa(\vartheta)$  as

$$\kappa(\vartheta) := \frac{\psi(\vartheta)\partial^2\psi(\vartheta)}{(\partial\psi(\vartheta))^2} = \mathbb{D}^2\psi(\vartheta). \quad (8.2.4)$$

Following Jacobsen et al. (2016, Section 3.3.3), we interpret  $\kappa(\vartheta)$  as the limiting ratio of the average excess degree of a susceptible node chosen randomly as a neighbour of an infectious individual, to the average degree of a susceptible node,  $\mu_S$ . The quantity  $\kappa(\vartheta)$  allows us to count various pairs accurately. In general, the operator  $\mathbb{D}^{r+1}\psi(\vartheta)$  recursively compares a susceptible node randomly chosen as a neighbour of  $r$  infected individuals with a randomly chosen susceptible node. Therefore, it allows us to count various  $r$ -configurations (triples, quadruples etc.) accurately in the limit. To be precise, in Lemma 8.5.1 in Section 8.5.1, we explicitly show

$$\mathbb{D}^{r+1}\psi(\theta) = \frac{\mu_S^{(r)}(\theta)}{\mu_S(\theta)} \mathbb{D}^r\psi(\theta) \xrightarrow{P} \mathbb{D}^{r+1}\psi(\vartheta), \quad (8.2.5)$$

where  $\mu_S^{(r)}$  is the average excess degree of a susceptible node randomly chosen as a neighbour of  $r$  infected individuals. In Section 8.5.1, we calculate these quantities explicitly.

Let us also define the operator  $\mathbb{H}(x, \vartheta) := (\mathbb{H}_x(x, \vartheta), \mathbb{H}_\vartheta(x, \vartheta))$ , where  $\mathbb{H}_x(x, \vartheta) := (\mathbb{H}_S(x_{SI}), \mathbb{H}_{SI}(x_S, x_{SI}, x_{SS}, \vartheta), \mathbb{H}_{SS}(x_S, x_{SI}, x_{SS}, \vartheta))$ , and  $\mathbb{H}_\vartheta(x_{SI}, \vartheta)$  are given by

$$\begin{aligned}\mathbb{H}_S(x_{SI}) &:= -\beta x_{SI}, \\ \mathbb{H}_{SI}(x_S, x_{SI}, x_{SS}, \vartheta) &:= \beta \kappa(\vartheta) \frac{x_{SI}}{x_S} (x_{SS} - x_{SI}) - \beta x_{SI}, \\ \mathbb{H}_{SS}(x_S, x_{SI}, x_{SS}, \vartheta) &:= -2\beta \kappa(\vartheta) \frac{x_{SI} x_{SS}}{x_S}, \\ \mathbb{H}_\vartheta(x_{SI}, \vartheta) &:= -\beta \frac{x_{SI}}{\alpha_S \partial \psi(\vartheta)}.\end{aligned}\tag{8.2.6}$$

Now, noting that A3 implies  $\sum_k k^2 p_k < \infty$ , recall the strong law on large graphs due to Jacobsen et al. (2016).

**Theorem 8.2.1** (Law of large numbers). *Assume A1, A2, and A3 for a configuration model graph  $\mathcal{G}(\psi, n)$ . Then, for any  $T > 0$ , the following holds*

$$\sup_{0 < t \leq T} \|(X(t)/n, \vartheta(t)) - (x, \vartheta)\| \xrightarrow{P} 0,$$

where  $(x, \vartheta) := ((x_S, x_{SI}, x_{SS}), \vartheta)$  is the solution of

$$(x(t), \vartheta(t)) = (x(0), \vartheta(0)) + \int_0^t \mathbb{H}(x(s), \vartheta(s)) \, ds,\tag{8.2.7}$$

with the initial condition  $x(0) = \alpha$  and  $\vartheta(0) = 1$ .

*Proof.* Observe that in the absence of recovery, the numbers of susceptible and infected individuals are linearly related as  $X_S + X_I = n$  in the standard Susceptible-Infected-Recovered (SIR) model. The proof therefore follows immediately by setting the recovery rate in the SIR model to zero and assuming that there is only one layer in Jacobsen et al. (2016). The crucial observation is that the neighbourhood distribution of a susceptible node, conditional on the process history, can be expressed as a hypergeometric distribution (see Remark 8.3.1) whose mixed moments can be approximated by the corresponding multinomial moments. This allows us to “average out” the individual-based quantities such as  $X_{SI,i}$  for  $i \in S$ . The convergence is then established by calculating several quadratic variations. The proof of our FCLT presented in Section 8.3 exploits similar calculations.  $\square$

### 8.3 FUNCTIONAL CENTRAL LIMIT THEOREM

In this section, we derive FCLT for  $X$  after an appropriate scaling. Define

$$M(t) = (M_S(t), M_{SI}(t), M_{SS}(t)) := n^{-1/2} M'(t).\tag{8.3.1}$$

We study the quadratic variation of the scaled martingale  $M(t)$ . The idea is to check whether either the optional or the predictable quadratic variation of the scaled process

$M$  converges in probability to a deterministic limit. If either of them does, and if the paths of  $M$  become approximately continuous in the limit (“big” jumps disappear), we can make use of the *Rebolledo’s theorem* (Helland 1982; Rebolledo 1980) to establish the asymptotic limit.

Note that  $M(t)$  is square integrable. For each  $\epsilon > 0$ , define

$$M^\epsilon(t) := (M_S^\epsilon(t), M_{SI}^\epsilon(t), M_{SS}^\epsilon(t)) \quad (8.3.2)$$

to be a vector of square integrable martingales containing all jumps of  $M(t)$  larger in absolute value than  $\epsilon$ . Define  $\mathcal{F}_{t-} := \sigma(\cup_{s \in [0, t)} \mathcal{F}_s)$ , the  $\sigma$ -field of events strictly prior to  $t \in \mathbb{R}_+$ . We write  $\mathcal{F}_{0-} := \mathcal{F}_0$ , by convention.

We use the shorthand notation  $\langle M \rangle(t)$  for the  $3 \times 3$  matrix of predictable covariation processes of the components of  $M(t)$ . That is,

$$\langle M \rangle(t) := \begin{pmatrix} \langle M_S \rangle(t) & \langle M_S, M_{SI} \rangle(t) & \langle M_S, M_{SS} \rangle(t) \\ \langle M_{SI}, M_S \rangle(t) & \langle M_{SI} \rangle(t) & \langle M_{SI}, M_{SS} \rangle(t) \\ \langle M_{SS}, M_S \rangle(t) & \langle M_{SS}, M_{SI} \rangle(t) & \langle M_{SS} \rangle(t) \end{pmatrix}. \quad (8.3.3)$$

Here  $\langle M_S \rangle(t) := \langle M_S, M_S \rangle(t)$  etc., by convention. Define  $\langle M^\epsilon \rangle$  similarly. We shall study the asymptotic limits of  $\langle M \rangle(t)$  and  $\langle M^\epsilon \rangle(t)$  as  $n \rightarrow \infty$  for each  $t \in \mathcal{T}$ . For this purpose, we need the neighbourhood distribution of a susceptible node  $i$  of degree  $k$ , i.e., the distribution of  $(X_{SI,i}, X_{SS,i})$  for a node  $i \in S_k$ , for all  $k \in \mathbb{N}$ .

**DYNAMIC CONSTRUCTION OF THE GRAPH** We make use of the dynamic graph construction method to derive the necessary probability distribution conditional on the history of the process. In this equivalent construction (Decreusefond et al. 2012; Jacobsen et al. 2016; Janson, M. Luczak, and Windridge 2014), the graph is dynamically revealed as infections take place. Accordingly, a susceptible node  $i \in S_k$  remains unpaired until it becomes infected. We could, however, pair off all unpaired edges at time  $t > 0$  (uniformly at random according to the CM construction) in order to define the neighbourhood of  $i$ . Therefore, the neighbourhood of a susceptible node arises solely out of uniform matching of half-edges. As a consequence, we obtain a hypergeometric distribution (see also Jacobsen et al. (2016)).

**Remark 8.3.1.** For  $k \in \mathbb{N}$  and  $i \in S_k$ , conditionally on the process history upto time  $t-$ , the vector  $(X_{SI,i}, X_{SS,i})$  follows a hypergeometric distribution

$$P(X_{SI,i} = n_{SI}, X_{SS,i} = n_{SS} \mid \mathcal{F}_{t-}) = \frac{\binom{X_{SI}}{n_{SI}} \binom{X_{S\bullet} - X_{SI}}{n_{SS}}}{\binom{X_{S\bullet}}{k}}, \quad (8.3.4)$$

supported on  $n_{SI} + n_{SS} = k$  where  $n_{SI}, n_{SS} \in \mathbb{N}_0$ .

This construction is equivalent in the sense that all the quantities of importance such as the number of susceptible nodes at time  $t \geq 0$  follow the same probability law as if the random multigraph was revealed first (according to uniform matching/pairing procedure of the configuration model) and then the SI epidemic process was run on it. We quote another important remark from Jacobsen et al. (2016) that would come in handy for the derivations.



**Remark 8.3.2.** Note that the total number of edges in the graph is  $2^{-1} \sum_i d_i$ . It immediately follows that  $n^{-1} X_{SI} \leq n^{-1} X_{S\bullet} \leq 2\partial\psi(1)$  and  $n^{-1} X_{SS} \leq n^{-1} X_{S\bullet} \leq 2\partial\psi(1)$  for sufficiently large  $n \in \mathbb{N}$ . Also note that  $\theta$  is a fractional quantity and therefore,  $\alpha_S \theta \partial\psi(\theta) \leq \partial\psi(1)$ . By virtue of [A1](#),  $n^{-1} X_{S\bullet}$  is bounded away from 0 on  $\mathcal{T}$  and hence, so is  $\theta$ . As a consequence of Jacobsen et al. ([2016](#), Lemma 1(b)), we can take the same lower bound for  $\alpha_S \theta \partial\psi(\theta)$ . Let us denote by  $\zeta > 0$  the uniform lower bound for  $n^{-1} X_{S\bullet}$  and  $\alpha_S \theta \partial\psi(\theta)$  so that we can write  $n^{-1} X_{S\bullet} \in [\zeta, 2\psi(1)] \subset \mathbb{R}_+$ .

### 8.3.1 Deterministic limit of $\langle M \rangle(t)$

Recall that  $(x, \vartheta) := ((x_S, x_{SI}, x_{SS}), \vartheta)$ . Let us begin by defining the following operators,

$$\begin{aligned}
 v_S &:= \beta x_{SI}, \\
 v_{SI} &:= \beta \left( \frac{x_{SI}(x_{SS} - x_{SI})^2}{x_S^2} \mathbb{D}^3 \psi(\vartheta) - \frac{x_{SI}(x_{SS} - 3x_{SI})}{x_S} \mathbb{D}^2 \psi(\vartheta) + x_{SI} \right), \\
 v_{SS} &:= 4\beta \frac{x_{SI} x_{SS}}{x_S} \left( \frac{x_{SS}}{x_S} \mathbb{D}^3 \psi(\vartheta) + \mathbb{D}^2 \psi(\vartheta) \right), \\
 v_{S,SI} &:= -\beta \left( \frac{x_{SI}(x_{SS} - x_{SI})}{x_S} \mathbb{D}^2 \psi(\vartheta) - x_{SI} \right), \\
 v_{S,SS} &:= 2\beta \frac{x_{SI} x_{SS}}{x_S} \mathbb{D}^2 \psi(\vartheta), \\
 v_{SI,SS} &:= -2\beta \frac{x_{SI} x_{SS}(x_{SS} - x_{SI})}{x_S^2} \mathbb{D}^3 \psi(\vartheta).
 \end{aligned} \tag{8.3.5}$$

Now, define a  $\mathcal{T}_0$ -indexed family of matrices  $\{V(t)\}$  as follows

$$V(t) := \begin{pmatrix} V_S(t) & V_{S,SI}(t) & V_{S,SS}(t) \\ V_{SI,S}(t) & V_{SI}(t) & V_{SI,SS}(t) \\ V_{SS,S}(t) & V_{SS,SI}(t) & V_{SS}(t) \end{pmatrix}, \tag{8.3.6}$$

where, given  $v_{id_1, id_2}(x, \vartheta)$  for  $id_1, id_2 \in \{S, SI, SS\}$  in [\(8.3.5\)](#),

$$V_{id_1, id_2}(t) := \int_0^t v_{id_1, id_2}(x(s), \vartheta(s)) \, ds, \tag{8.3.7}$$

with the convention  $v_{id_1, id_2} := v_{id_2, id_1}$  for  $id_1, id_2 \in \{S, SI, SS\}$  and  $v_{id_1, id_2} := v_{id_1}$  whenever  $id_1 = id_2 \in \{S, SI, SS\}$ . This also sets the convention  $V_{id_1, id_2}(t) := V_{id_2, id_1}(t)$  for  $id_1, id_2 \in \{S, SI, SS\}$  and  $V_{id_1, id_2}(t) := V_{id_1}(t)$  whenever  $id_1 = id_2 \in \{S, SI, SS\}$  for each  $t \in \mathcal{T}_0$ .

Let us now present our first result providing the deterministic limit of  $\langle M \rangle$  in the following lemma. The key strategy in proving these deterministic limits will be to approximate various hypergeometric moments by the corresponding multinomial moments. The proof of Lemma [8.3.1](#) is, however, lengthy and somewhat involved. Therefore, it is provided in Appendix [F.2](#).

**Lemma 8.3.1** (Deterministic limit of  $\langle M \rangle$ ). *Consider the stochastic SI model described in Section 8.1. Assume A1, A2 and A3 for a configuration model graph  $\mathcal{G}(\psi, n)$ . Then,*

$$\langle M \rangle(t) \xrightarrow{P} V(t),$$

for each  $t \in \mathcal{T}_0$ , as  $n \rightarrow \infty$  where  $V(t)$  is as defined in (8.3.6), and  $(x, \vartheta)$  is the solution of (8.2.7) with  $x(0) = \alpha$  and  $\vartheta(0) = 1$ .

### 8.3.2 Asymptotic rarefaction of jumps

Recall that  $M^\epsilon := (M_S^\epsilon, M_{SI}^\epsilon, M_{SS}^\epsilon)$  is the vector of square integrable martingales containing all jumps of components of  $M$  larger than  $\epsilon$  in absolute value, for  $\epsilon > 0$ , i.e.,  $M_{id}(t) - M_{id}^\epsilon(t)$  is a local square integrable martingale and  $|\delta M_{id}(t) - \delta M_{id}^\epsilon(t)| \leq \epsilon$  for all  $id \in \{S, SI, SS\}$  and  $t \in \mathcal{T}_0$ . We wish to show  $\langle M_{id}^\epsilon \rangle(t) \xrightarrow{P} 0$  for all  $id \in \{S, SI, SS\}$  and  $t \in \mathcal{T}_0$ , as  $n \rightarrow \infty$ . We would like to point out that this condition is essentially the *strong Asymptotic Rarefaction of Jumps Condition of the second type* (strong ARJ(2)) as described in Andersen et al. (1993) and Rebolledo (1980). Intuitively this ensures that the sample paths of the martingale  $M(t)$  are close to continuous in the limit. Before proceeding further, we offer the following remark.

**Remark 8.3.3** (Limit of the maximum degree). For the configuration model graph  $\mathcal{G}(\psi, n)$  along with A3, the following holds true:

$$n^{-\frac{1}{2}} d_{\max} \xrightarrow{\text{a.s.}} 0, \quad (8.3.8)$$

where  $d_{\max}$  is the maximum degree observed in a realisation of  $\mathcal{G}(\psi, n)$ .

*Proof of Remark 8.3.3.* The result follows by a direct application of the result in Barndorff-Nielsen (1963, Theorem 5.2) along with A3. □

Let us now compute the predictable quadratic variation of  $M^\epsilon$  and establish its asymptotic limit.

**Lemma 8.3.2** (Limit of  $\langle M_{id}^\epsilon \rangle$ ). *Consider the stochastic SI model described in Section 8.1. Assume A1, A2 and A3 for a configuration model graph  $\mathcal{G}(\psi, n)$ . Consider the vector  $M^\epsilon$  of square integrable martingales containing all jumps of components of  $M(t)$  larger than  $\epsilon$  in absolute value for  $\epsilon > 0$ , as defined in (8.3.2). Then, as  $n \rightarrow \infty$ , for all  $id \in \{S, SI, SS\}$ , for each  $t \in \mathcal{T}_0$ ,*

$$\langle M_{id}^\epsilon \rangle(t) \xrightarrow{P} 0. \quad (8.3.9)$$

*Proof of Lemma 8.3.2.* We proceed in the following two steps.

COMPUTATION OF  $\langle M_S^\epsilon \rangle$  Note that the original process  $M'_S$  makes only unit jumps. Then, for arbitrary  $\epsilon > 0$ ,

$$\begin{aligned} \langle M_S^\epsilon \rangle(t) &\leq \int_0^t \mathbb{E}[(\delta M_S^\epsilon(s))^2 \mathbb{1}(|\delta M'_S(s)| > n^{1/2}\epsilon) \mid \mathcal{F}_{s-}] ds = 0 \quad \forall n > \frac{1}{\epsilon^2} \\ \implies \langle M_S^\epsilon \rangle(t) &\xrightarrow{P} 0 \text{ for all } 0 < t \leq T \text{ and for all } \epsilon > 0 \text{ as } n \rightarrow \infty. \end{aligned}$$

COMPUTATION OF  $\langle M_{SI}^\epsilon \rangle$  AND  $\langle M_{SS}^\epsilon \rangle$  Note that both  $M'_{SI}$  and  $M'_{SS}$  jump only if infection of a node occurs. This in particular implies that the jump sizes of  $M'_{SI}$  and  $M'_{SS}$  are bounded above by the degree of the node getting infected. Therefore, they are also bounded above by the maximum degree  $d_{max}$ . For an arbitrary  $\epsilon > 0$ , and for  $id \in \{SI, SS\}$ ,

$$\begin{aligned} \langle M_{id}^\epsilon \rangle(t) &\leq \int_0^t \mathbb{E}[(\delta M_{id}^\epsilon(s))^2 \mathbb{1}(|n^{-1/2}d_{max}| > \epsilon) \mid \mathcal{F}_{s-}] ds \\ &\leq tn^{-1}d_{max}^2 \mathbb{1}(|n^{-1/2}d_{max}| > \epsilon). \end{aligned}$$

By Remark 8.3.3, and the continuous mapping theorem as well as standard properties of convergence in probability, the right hand side of the above inequality

$$tn^{-1}d_{max}^2 \mathbb{1}(|n^{-1/2}d_{max}| > \epsilon) \xrightarrow{P} 0$$

for each  $0 < t \leq T$  and  $\epsilon > 0$ . Therefore, for all  $\delta > 0$ , we have  $P(\langle M_{id}^\epsilon \rangle(t) > \delta) \leq P(tn^{-1}d_{max}^2 \mathbb{1}(|n^{-1/2}d_{max}| > \epsilon) > \delta) \rightarrow 0$  as  $n \rightarrow \infty$ , establishing  $\langle M_{id}^\epsilon \rangle(t) \xrightarrow{P} 0$  as  $n \rightarrow \infty$  for all  $0 < t \leq T$  and  $\epsilon > 0$ . This completes the proof.  $\square$

### 8.3.3 Statement and proof of the FCLT

Having shown the convergence of all relevant quadratic variation processes, we are now ready to present the following FCLT. First we state that the function  $V$  found in Lemma 8.3.1 is a positive semi-definite (psd) matrix-valued function on  $\mathcal{T}_0$ , with psd increments. Set  $V(0) := \mathbf{0}$ , the  $3 \times 3$  null matrix, so that we can treat  $V(t)$  as a psd matrix-valued function on the entirety of  $\mathcal{T}$ . Let us denote the collection of all such psd  $3 \times 3$  matrix-valued functions on  $\mathcal{T}$  that has psd increments and that is  $\mathbf{0}$  at time zero by  $\mathcal{V}$ . Given such a matrix-valued function  $V \in \mathcal{V}$ , let  $W$  be a continuous Gaussian vector martingale such that  $\langle W \rangle = [W] = V$ . Such a process always exists (Andersen et al. 1993, Chapter II, p. 83). In particular,  $W(t) - W(s) \sim N(\mathbf{0}, V(t) - V(s))$ , the multivariate normal distribution for  $0 \leq s \leq t$ .

**Theorem 8.3.1** (Functional Central Limit Theorem). *Consider the stochastic SI model described in Section 8.1. Assume A1, A2 and A3 for a configuration model graph  $\mathcal{G}(\psi, n)$ . Consider, for  $t \in \mathcal{T}$ , the fluctuation process*

$$Y(t) := \sqrt{n} \left( \frac{1}{n} X(t) - x(t) \right). \quad (8.3.10)$$

Assume  $\lim_{n \rightarrow \infty} Y(0) = U(0)$ , for some nonrandom  $U(0)$ . Then, there exists a matrix-valued function  $V \in \mathcal{V}$  on  $\mathcal{T}$  such that

$$Y \xrightarrow{\mathcal{D}} U \text{ in } D^{(3)} \text{ as } n \rightarrow \infty, \quad (8.3.11)$$

where  $U$  is a continuous Gaussian vector semimartingale satisfying

$$U(t) = U(0) + W(t) + \int_0^t \nabla \mathbb{H}_x(x(s), \vartheta(s)) U(s) ds, \quad (8.3.12)$$

where  $\nabla \mathbb{H}_x(x, \vartheta) := ((\partial_j \mathbb{H}_i(x, \vartheta)))$  for  $i, j \in \{S, SI, SS\}$  and  $W$  is a continuous Gaussian vector martingale such that  $\langle W \rangle = [W] = V$ , provided  $V$  remains finite on the entirety of  $\mathcal{T}$  and  $\nabla \mathbb{H}_x(x(s), \vartheta(s))$  is continuous.

Here  $D^{(3)}$  is the space of  $\mathbb{R}^3$ -valued càdlàg functions on  $\mathcal{T}$  endowed with the Skorohod topology and  $\xrightarrow{\mathcal{D}}$  stands for weak convergence.

*Proof of Theorem 8.3.1.* We first prove an FCLT for the martingale process  $M$  defined in (8.3.1). We wish to apply Rebolledo's FCLT for local martingales on  $M$ . Please refer to Rebolledo (1980) for the original version of the theorem and Andersen et al. (1993, Chapter II, p. 83) for a version tailored to locally square integrable martingales. Please note that in the light of Doob-Meyer decomposition given in (8.1.2),  $M$  is indeed a pure jump, zero-mean, locally square integrable, càdlàg martingale. After having established an FCLT for the martingale process  $M$ , we prove convergence of the fluctuation process  $Y$ . It suffices to carry out the following three steps.

(STEP I) DETERMINISTIC LIMIT OF  $\langle M \rangle$  Let  $(x, \vartheta)$  be the solution of (8.2.7) with initial condition  $x(0) = \alpha$  and  $\vartheta(0) = 1$ , as given in Theorem 8.2.1. Then, by virtue of Lemma 8.3.1, we conclude, for each  $t \in \mathcal{T}_0$ ,  $\langle M \rangle(t) \xrightarrow{P} V(t)$ , where the matrix-valued function  $V$  is defined in (8.3.6), and we set  $V(0) := \mathbf{0}$ , the  $3 \times 3$  null matrix.

(STEP II) ASYMPTOTIC RAREFACTION OF JUMPS Let  $\epsilon > 0$  be arbitrary. Consider the vector  $M^\epsilon$  of square integrable martingales containing all jumps of components of  $M(t)$  larger than  $\epsilon$  in absolute value for  $\epsilon > 0$ , as defined in (8.3.2). Then, by means of Lemma 8.3.2, we conclude  $\langle M_{id}^\epsilon \rangle(t) \xrightarrow{P} 0$ , for each  $t \in \mathcal{T}_0$  and  $id \in \{S, SI, SS\}$ .

Now let  $W$  be the continuous Gaussian vector martingale such that  $\langle W \rangle = [W] = V$ . In the light of Rebolledo's theorem for locally square integrable martingales (Andersen et al. 1993, Chapter II, p. 83), Step I and Step II are sufficient to establish

$$(M(t_1), M(t_2), \dots, M(t_l)) \xrightarrow{\mathcal{D}} (W(t_1), W(t_2), \dots, W(t_l)) \text{ as } n \rightarrow \infty$$

for all  $t_1, t_2, \dots, t_l \in \mathcal{T}_0$ . Furthermore, since  $\mathcal{T}_0$  is dense in  $\mathcal{T}$ , we conclude  $M \xrightarrow{\mathcal{D}} W$  in  $D^{(3)}$  as  $n \rightarrow \infty$ , and  $\langle M \rangle$  and  $[M]$  converge uniformly on compact subsets of  $\mathcal{T}$ , in probability, to  $V$ .

(STEP III) CONVERGENCE OF THE FLUCTUATION PROCESS In keeping with the Doob-Meyer decomposition given in (8.1.2),

$$Y(t) = Y(0) + M(t) + \int_0^t \sqrt{n} \left( \frac{1}{n} \mathbb{F}_X(X(s)) - \mathbb{H}_x(x(s), \vartheta(s)) \right) ds,$$

we expect the following limit process

$$U(t) = U(0) + W(t) + \int_0^t \nabla \mathbb{H}_x(x(s), \vartheta(s)) U(s) ds. \quad (8.3.13)$$

Indeed, define

$$\begin{aligned} \Delta(t) &:= \int_0^t \sqrt{n} \left( \frac{1}{n} \mathbb{F}_X(X(s)) - \mathbb{H}_x(x(s), \vartheta(s)) - \frac{1}{\sqrt{n}} \nabla \mathbb{H}_x(x(s), \vartheta(s)) Y(s) \right) ds \\ &= \int_0^t \sqrt{n} \left( \frac{1}{n} \mathbb{F}_X(X(s)) - \mathbb{H}_x\left(\frac{1}{n} X(s), \vartheta(s)\right) + \mathbb{H}_x\left(\frac{1}{n} X(s), \vartheta(s)\right) \right. \\ &\quad \left. - \mathbb{H}_x(x(s), \vartheta(s)) - \frac{1}{\sqrt{n}} \nabla \mathbb{H}_x(x(s), \vartheta(s)) Y(s) \right) ds. \end{aligned}$$

Note that the strong law of large numbers in Theorem 8.2.1 establishes uniform convergence (in probability) of the operators  $n^{-1} \mathbb{F}_X(X(s))$  and  $\mathbb{H}_x(\frac{1}{n} X(s), \vartheta(s))$ , and the latter operator is Lipschitz continuous on its domain (see Jacobsen et al. (2016)). In the light of Theorem 8.2.1 and A3, it follows from the Lipschitz continuity of various multinomial compensators  $C_m^k$  introduced in the proof of Lemma 8.3.1 that

$$\lim_{n \rightarrow \infty} \sqrt{n} \left( \frac{1}{n} \mathbb{F}_X(X(s)) - \mathbb{H}_x\left(\frac{1}{n} X(s), \vartheta(s)\right) \right) = 0.$$

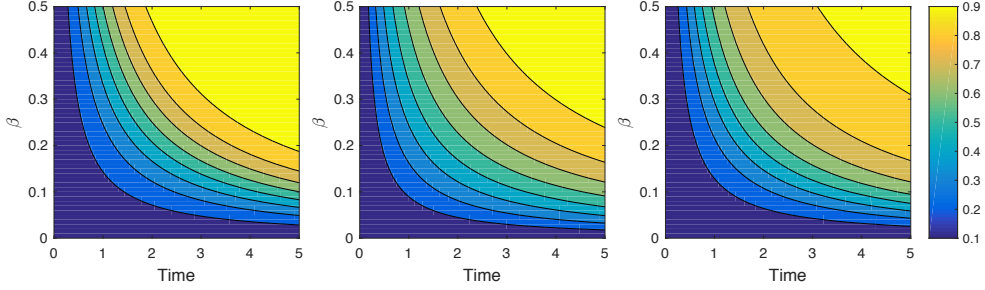
Moreover, we have just shown  $M \xrightarrow{\mathcal{D}} W$  in  $D^{(3)}$ . If  $V$  remains finite on the entirety of  $\mathcal{T}$ , the matrix-valued function  $\nabla \mathbb{H}_x(x(s), \vartheta(s))$  is continuous, and  $\lim_{n \rightarrow \infty} Y(0) = U(0)$ , for some nonrandom  $U(0)$ , then we have  $\sup_{t \in \mathcal{T}} |\Delta(t)| \xrightarrow{\mathbb{P}} 0$  following Theorem 8.2.1, and by application of the continuous mapping theorem, we conclude

$$Y \xrightarrow{\mathcal{D}} U \text{ in } D^{(3)} \text{ as } n \rightarrow \infty,$$

where the Gaussian semimartingale  $U$  satisfies (8.3.13) with the Gaussian martingale  $W$  being such that  $\langle W \rangle = [W] = V$ . This completes the proof.  $\square$

## 8.4 APPLICATIONS

Here, we consider some applications of our result. As we discuss these applications, we shall also present some numerical and simulation results that are intended not only to provide insights into the dynamics of the process, but also to serve as a verification of our results.

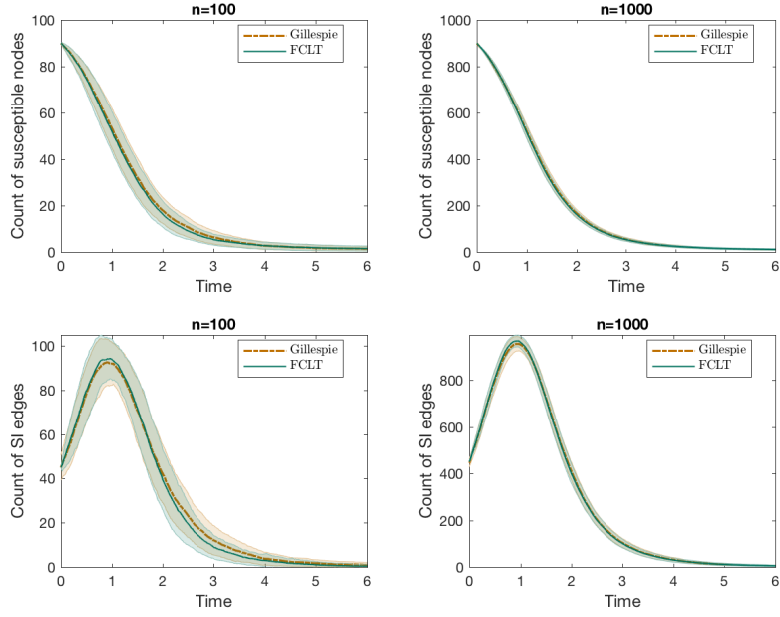


**Figure 8.2:** Comparison of percolation profiles of three degree distributions having the same mean. **(Left)** Poisson distribution with mean 6. **(Middle)** A heterogeneous population with degree distribution  $p_k := 0.7 \times \mathbb{1}(k=1) + 0.2 \times \mathbb{1}(k=4) + 0.1 \times \mathbb{1}(k=45)$ . Such a degree distribution represents a population segregated into three classes. Weak nodes constitute the biggest class, followed by medium strength nodes and then strong nodes. **(Right)** Negative Binomial distribution with parameters  $r=2, p=3/4$ . The figures show time evolution of the fraction of nodes on the percolated component for varying infection rates  $\beta$ . We assume the initial fraction of infected nodes is 0.1 in all three cases. The yellow region in each of the plots corresponds to the terminal state. Questions such as whether the system with an infection rate  $\beta$  “percolates” are immediately settled by drawing a horizontal line and checking whether the lines passes through the colour corresponding to a pre-specified level.

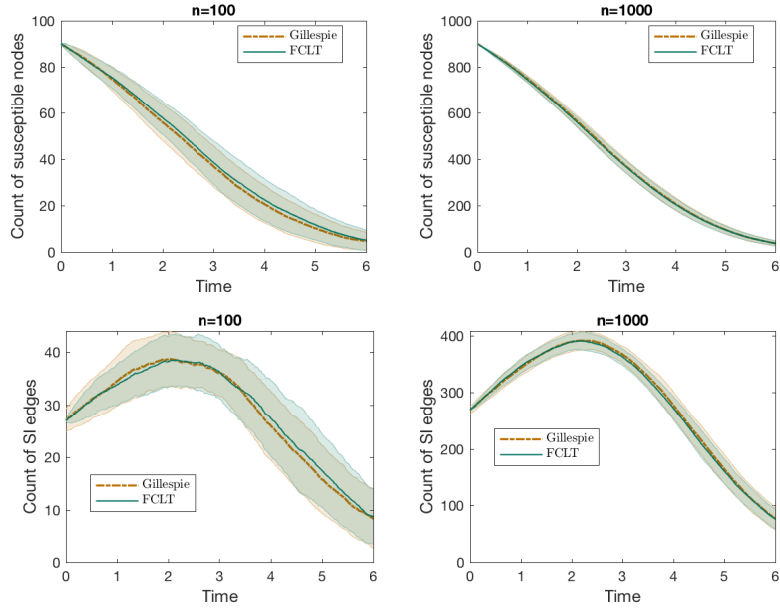
#### 8.4.1 Percolation

There is a connection between the stochastic SI model and percolation theory from statistical physics. It is the study of a liquid filtering (“percolating”) through a porous medium. Classical equilibrium-mechanics studies its stationary behaviour and premises upon the axiom that the underlying quantum-mechanical laws are designed so as to maximise the entropy. Stationary distribution of such a stochastic system is given by the Boltzmann ensemble. This classical treatment of the subject, however, does not explain the non-equilibrium behaviour of the dynamical system, *i.e.*, when it is still in a transient phase. Consequently the non-equilibrium behaviour of percolation has aroused much interest in recent times. Some notable contributions include Barato and Hinrichsen (2009) and Hinrichsen (2006). Standard treatment of percolation, both equilibrium and non-equilibrium, has been extended in another important direction concerning the structure of the porous medium. Traditionally it has been studied on lattices and grids. Of late, however, percolation on random graphs has also been considered (Baroni, Hofstad, and Komjathy 2015; Callaway et al. 2000; Hofstad 2010). Continuing in this direction, we shall treat (non-equilibrium) percolation as a dynamical process on a configuration model random graph and study its behaviour over a finite time interval.

One of the key quantities of interest in the study of non-equilibrium percolation is the time evolution of the number of wetted sites (also called “active” nodes in the literature). The correspondence of our stochastic SI model as described in Section 8.1 to non-equilibrium percolation is visible if we treat the infected nodes as the ones wetted

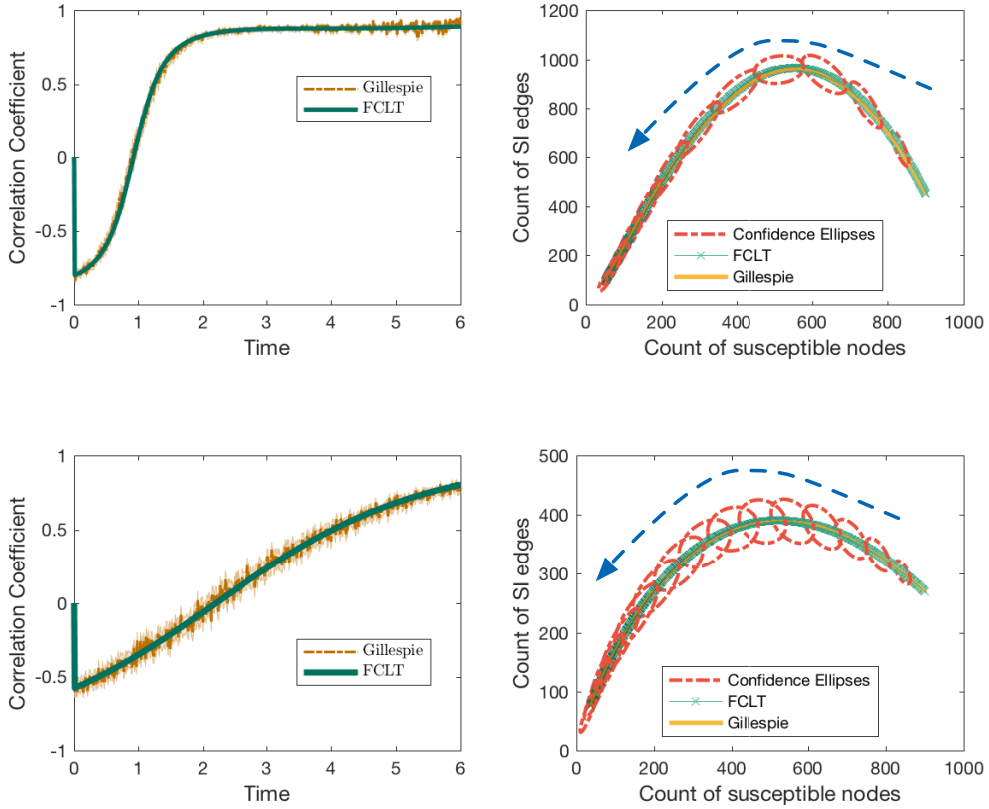


(a) Simulation setting: Poisson distribution with  $\lambda = 5$ ,  $\alpha_S = 0.9$ , and  $\beta = 0.5$ .



(b) Simulation setting:  $r$ -regular random graph with  $r = 3$ ,  $\alpha_S = 0.9$ , and  $\beta = 0.5$ .

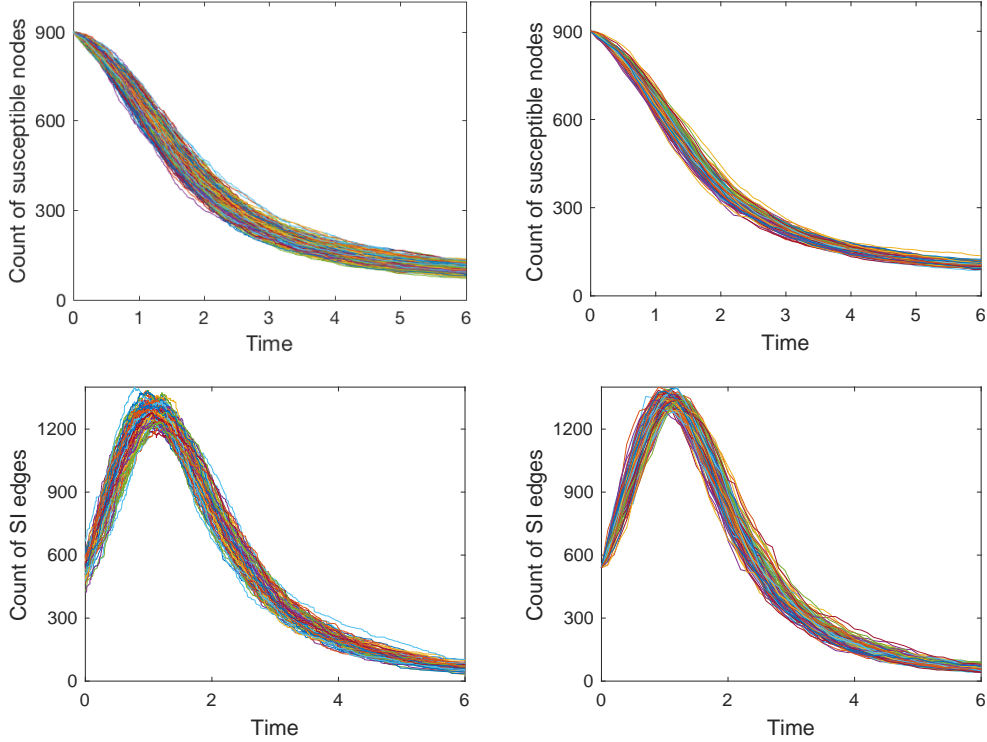
**Figure 8.3:** Comparison of our diffusion approximation with simulation results obtained by Gillespie's algorithm.



**Figure 8.4:** The figures on the left depict the time evolution of the correlation coefficient between jumps of  $X_S$  and  $X_{SI}$  as estimated from numerical simulations (via Gillespie's algorithm) pitted against theoretical values computed from the functional central limit theorem (Theorem 8.3.1). The figures on the right show the expected sample path in the space of  $X_S$  and  $X_{SI}$ . The two lines correspond to numerical simulation and theoretical values. The dotted ellipses are the 95%-confidence ellipses. The arrows indicate the time direction. **(Above)** Poisson distribution with mean 5. **(Below)**  $r$ -regular random graph with  $r = 3$ . In both cases,  $n = 1000$ ,  $\alpha_S = 0.9$ , and  $\beta = 0.5$ .

during the process of percolation. Accordingly, in this context, we give the process  $X(t)$  appropriate new interpretation. The process  $X_S(t)$ , for example, captures the number of unwetted sites until time  $t$ , and the process  $X_{SI}$ , the number of channels (bonds) through which the liquid can percolate. In Figure 8.1, the percolated component up to a given time (the wetted part of the graph) is shown in red. Having made the correspondence precise, we can apply Theorem 8.3.1 to approximate these quantities in the large graph limit.

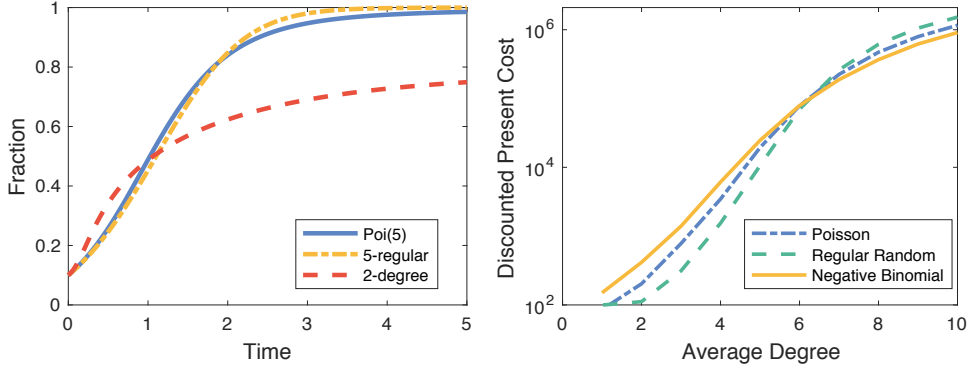




**Figure 8.5:** Comparison of simulated sample paths. **(Left)** Sample paths obtained through Gillespie's algorithm. **(Right)** Sample paths obtained through diffusion approximation. Simulation setting:  $n = 1000$ , Negative Binomial distribution with  $r = 2, p = 3/4$ .

**NUMERICAL ILLUSTRATION** In Figure 8.3, we show some simulation results to check the accuracy of our scaling limit. We compare the expected sample paths of  $X_S$  and  $X_{SI}$  provided by Theorem 8.2.1 and Theorem 8.3.1, with estimates obtained using simulations of the Gillespie's algorithm on a CM graph. In particular, we considered a Poisson degree distribution in Figure 8.3a and a 3-regular random graph in Figure 8.3b (obtained by the CM construction with degree distribution  $p_k = \mathbb{1}(k = 3)$ ). In Figure 8.4, we show the time evolution of the correlation coefficient between the jumps of  $X_S$  and  $X_{SI}$ , and also the expected sample path coupled with 95%-confidence ellipses in the space of  $X_S$  and  $X_{SI}$ .

On a slightly different note, the existence of a giant component and the proportion of nodes on the giant component play an important role in percolation theory, especially from an equilibrium point of view in statistical mechanics. The case of a degree distribution  $\{p_k\}_{k \in \mathbb{N}_0}$  such that  $\sum_{k \in \mathbb{N}_0} k^2 p_k = 2 \sum_{k \in \mathbb{N}_0} k p_k$  and  $p_1 = 0$  is a curious one in that quite different behaviours of the giant component are observable for such a degree distribution. Please refer to Janson and M. J. Luczak (2009) for examples of such behaviours. Barring this exceptional case, in the light of A3, the condition for existence of a giant component is satisfied (see Molloy and Reed (1995)) for our stochastic SI model



**Figure 8.6: (Left)** Comparison of the time evolution of the fraction of nodes on the percolated component for different degree distributions with the same mean. The 2-degree distribution in the plot refers to the degree distribution  $p_k = 0.5 \times \mathbb{1}(k = 1) + 0.5 \times \mathbb{1}(k = 9)$ , where none of the nodes have degree 5 yet the average degree is 5. This presents a pathological case and highlights the need to take into account higher moments of the degree distribution. **(Right)** Comparison of discounted cost against increasing average degree for different degree distributions. With increasing average degree the graphs lose sparsity and facilitate spread of computer virus. Therefore they incur higher cost. When the average degree is very small, regular random graphs seem favourable compared to random graphs with negative binomial distributed degrees. The costs are computed with  $n = 1000, \gamma = 1$ .

in the traditional sense. To be precise, setting  $\alpha_S = 1$  and taking asymptotic limit in time, one finds the fraction of nodes on the giant component to be  $1 - \psi(\theta_\infty)$ , where  $\theta_\infty > 0$  is the solution of  $\partial\psi(1)\theta_\infty = \partial\psi(\theta_\infty)$  (see Janson, M. Luczak, and Windridge (2014) and Molloy and Reed (1998)). However, as mentioned earlier, we take a non-equilibrium point of view and concern ourselves with the time evolution of the fraction of nodes on the infected part of the graph, the “percolated component”. As a by-product of the scaling limits in Theorem 8.2.1 and Theorem 8.3.1, the variable  $\theta$  defined in (8.1.4) gives us a tool to approximate the proportion of susceptible individuals in the population (and hence, the proportion of infected nodes as well). We expect the fraction of infected individuals to converge in probability to  $1 - \alpha_S\psi(\theta)$  as  $n \rightarrow \infty$ ,  $\theta$  being the scaling limit of  $\theta$ . A fixed time interval  $\mathcal{T}$  enables us to look for critical values in the space of the infection rate  $\beta > 0$ . This allows us to decide whether the system “percolates” in the sense that the fraction of nodes on the percolated component achieves a value greater than a pre-specified one (usually close to unity) by time  $T$ . Using different colours in Figure 8.2, we depict the fraction of nodes on the percolated component as a function of both time and the infection rate  $\beta$  (let us call such a figure a percolation profile) for three degree distributions with the same mean. Questions such as whether the system with an infection rate  $\beta$  “percolates” are immediately settled by drawing a horizontal line and checking whether the line passes through the colour corresponding to the pre-specified

level. See Figure 8.6 for another comparative view, highlighting the need to take into account higher moments of the degree distribution.

#### 8.4.2 The strange case of Poisson-type degree distributions and the exactness of the pair approximation

We consider a particular class of degree distributions called Poisson-type (PT) by Jacobsen et al. (2016). A degree distribution with PGF  $\psi$  is called PT if  $\kappa(\vartheta)$ , defined in (8.2.4), is a constant, i.e.,  $\kappa(\vartheta) = \kappa$  for some  $\kappa \in \mathbb{R}$  (or equivalently,  $\partial\psi(\vartheta) = \partial\psi(1) (\psi(\vartheta))^\kappa$ ). As a consequence, the operators defined in (8.2.3) are also constants, and satisfy

$$\mathbb{D}^r \psi(\vartheta) = \prod_{i=1}^{r-1} (i\kappa - i + 1) = [(r-1)\kappa - r + 2] \mathbb{D}^{r-1} \psi(\vartheta),$$

with  $\mathbb{D}^0 \psi(\vartheta) = 1$ . The PT class includes Poisson ( $\kappa = 1$ , irrespective of the mean of the distribution), degenerate distribution ( $r$ -regular random graphs,  $\kappa = \frac{r-1}{r} < 1$ ), binomial ( $\kappa = \frac{N-1}{N} < 1$ , independent of  $p$  for  $\text{Bin}(N, p)$ ), negative binomial ( $\kappa = \frac{r+1}{r} > 1$ , independent of  $p$  for  $\text{NB}(r, p)$ ) degree distributions. The PT class is particularly peculiar in that it totally decouples the vector  $x = (x_S, x_{SI}, x_{SS})$ , and the matrix-valued function  $V$  from the auxiliary variable  $\vartheta$  so that an autonomous system of ODEs can be obtained for  $x$  and  $V$ , rendering  $\vartheta$  redundant. This allows for great simplification in the limiting equations. Define  $\mathbb{G}(x) := (\mathbb{G}_S(x), \mathbb{G}_{SI}(x), \mathbb{G}_{SS}(x))$  as

$$\begin{aligned} \mathbb{G}_S(x) &:= -\beta x_{SI}, \\ \mathbb{G}_{SI}(x) &:= \beta \kappa \frac{x_{SI}}{x_S} (x_{SS} - x_{SI}) - \beta x_{SI}, \\ \mathbb{G}_{SS}(x) &:= -2\beta \kappa \frac{x_{SI} x_{SS}}{x_S}. \end{aligned} \tag{8.4.1}$$

Plugging  $\mathbb{D}^2 \psi(\vartheta) = \kappa$ , and  $\mathbb{D}^3 \psi(\vartheta) = \kappa(2\kappa - 1)$  in (8.3.5), the matrix-valued function  $V$  is entirely determined by  $x$ . The following is immediate.

**Corollary 8.4.1** (Scaling limit under PT distributions). *Assume A1, A2, and A3 for a configuration model graph  $\mathcal{G}(\psi, n)$  with  $\partial\psi(\vartheta) = \partial\psi(1) (\psi(\vartheta))^\kappa$  for some  $\kappa \in \mathbb{R}$ . Then, the following law of large numbers holds*

$$\sup_{0 \leq t \leq T} \|n^{-1} X(t) - x(t)\| \xrightarrow{P} 0,$$

where  $x$  is the solution of  $x(t) = x(0) + \int_0^t \mathbb{G}(x(s)) ds$  with  $x(0) = \alpha$ . Moreover, the fluctuation process  $Y$  defined in (8.3.10) converges weakly to a continuous Gaussian vector semimartingale  $U$  satisfying

$$U(t) = U(0) + W(t) + \int_0^t \nabla \mathbb{G}(x(s)) U(s) ds,$$

where  $W$  is a Gaussian vector martingale such that  $\langle W \rangle = [W] = V$ .

In fact, one can obtain a smaller system by expressing  $x_{SI}$  and  $x_{SS}$  explicitly as a function of  $x_S$  (see Jacobsen et al. (2016)). This is remarkable because, under the PT class, the graph structure impacts the scaling limits *only* through two summary statistics of  $\psi$ , namely the mean  $\partial\psi(1)$  and  $\kappa = \mathbb{D}^2\psi(1)$ . Recall that  $\kappa$  is the limiting ratio of the average excess degree of a susceptible node chosen at random as a neighbour of an infected node, to the average degree of a susceptible node. Under the PT class, this ratio remains constant throughout the entire course of time  $\mathcal{T}$ . The mean  $\partial\psi(1)$  only impacts the initial condition  $x(0) = \alpha$  through (8.2.2). The dynamics of the limiting process are then dictated by  $\kappa$ .

Now we revisit the correlation equations approach of Rand (2009) from ecology literature to study the dynamics of counts of singles, pairs, triples, and quadruples of the form  $A, AB, ABC, ABCD$ , where  $A, B, C, D \in \{S, I\}$ . Following Rand (2009), we use the notation  $[\cdot]$  to denote the count. In this mean-field approach, the dynamics of singles are described by that of pairs; dynamics of pairs, by triples, and so on. In this context, pair approximation refers to approximating the count of triples by pairs in the following way

$$[ABC] \approx \kappa \frac{[AB][BC]}{[B]},$$

and closing the system at the level of pairs (also known as pair closure). In order to draw an analogy, we divide the counts by  $n$ , and use the same notation for the scaled counts. We also set the same initial condition  $([S], [SI], [SS]) = \alpha$  at  $t = 0$ . The pair approximation then yields a system of ODEs for  $([S], [SI], [SS])$  that exactly matches the limiting ODEs for  $n^{-1}X$ , i.e.,

$$\frac{d}{dt}([S], [SI], [SS]) = \mathbb{G}([S], [SI], [SS]). \quad (8.4.2)$$

Therefore, under the PT class, the pair approximation is *exact* in the sense that it correctly estimates the limiting means of various counts. By virtue of Corollary 8.4.1, our FCLT further enables it to correctly estimate *all* other higher limiting moments, because  $V$  is now entirely determined by the solution of (8.4.2). As the PT class is quite big, our FCLT thus greatly enhances the usefulness of the pair approximation.

### 8.4.3 Spread of Computer Viruses

Epidemic models have been used in the context of spread of computer virus for some years now. The correspondence between our model and the application area under consideration is apparent without requiring much change in nomenclature. Early works in this direction did not take into account the inherent graph structure and assumed “homogeneous mixing” in some sense. Recent works, however, duly studied it on realistic computer networks, often modelled as random graphs. Lelarge, for example, based much of his work on classical Erdős-Rényi (ER) random graphs and CMs (see, e.g., Lelarge (2012)). Interested readers are referred to Kephart and White (1993), Lelarge (2012), and Wierman and Marchette (2004) for an overview of relevant literature. Applying our results, we can approximate the number of virus-affected computers over time and the edges of different types. Additionally, one might be interested in estimating

some “cost” involving the count variables in a linear or non-linear fashion. For instance, if the cost function is polynomial in the count variables, the mixed moments of various orders can be approximated by means of Theorem 8.3.1. To illustrate the concept using a simple example, we assume an exponentiated form for the incurred cost to emphasise the severity of a computer being virus-affected. We can then compute time-discounted expected incurred cost and study how it behaves with decreasing sparsity of the underlying graph. To this end, define

$$\begin{aligned} I(t) &:= \exp(cX_I(t)), \\ C_\psi &:= \mathbb{E} \left[ \int_{\mathcal{T}} \exp(-\gamma t) I(t) dt \right] = \int_{\mathcal{T}} \exp(-\gamma t) \mathbb{E}[I(t)] dt, \end{aligned} \quad (8.4.3)$$

where  $c > 0$  and  $\gamma > 0$  are constants. In Figure 8.6, we plot the discounted cost  $C_\psi$  against an increasing average degree of the underlying graph, engendering decreasing sparsity. When the average degree is very small, regular random graphs seem favourable compared to random graphs with negative binomial distributed degrees.

## 8.5 DISCUSSION

### 8.5.1 Interpretation of the $\mathbb{D}$ operator

Here, we provide an intuitive explanation for the  $\mathbb{D}$  operator defined in (8.2.3) in the context of SI process on CM random graphs. Recall that  $\mu_S$ , and  $\mu_S^{(r)}$  denote the average degree of a randomly chosen susceptible node, and the average excess degree of a susceptible node randomly chosen as a neighbour of  $r$  infected individuals, respectively. In Section 8.2, we mentioned the operator  $\mathbb{D}^{r+1}\psi(\theta)$  recursively compared a susceptible node randomly chosen as a neighbour of  $r$  infected individuals with a randomly chosen susceptible node. We make this notion of comparison precise.

**Lemma 8.5.1** (Interpretation of the  $\mathbb{D}$  operator). *Assume A1, A2, and A3 for the stochastic SI model on configuration model graph  $\mathcal{G}(\psi, n)$ . Then,  $\mathbb{D}^r\psi(\theta) \xrightarrow{P} \mathbb{D}^r\psi(\theta)$  uniformly on  $\mathcal{T}$ , and the following recurrence relation for  $\mathbb{D}^r$  holds*

$$\mathbb{D}^{r+1}\psi(\theta) = \frac{\mu_S^{(r)}(\theta)}{\mu_S(\theta)} \mathbb{D}^r\psi(\theta). \quad (8.5.1)$$

The proof of Lemma 8.5.1 is provided in Appendix F.3.

### 8.5.2 Extension to SIR epidemic processes

In our present work, we have disregarded “recovery” of the infected nodes. The reason behind this exclusion is our inability to evaluate the neighbourhood distribution of an infected node in the presence of spontaneous recovery of its neighbours. One difficulty is that, unlike the susceptible nodes (of a given degree) that are untouched by the process of infection and hence, receive identically distributed neighbourhoods upon uniformly random matching of half-edges, the infected nodes are not identically distributed because they already possess partially formed neighbourhoods consisting of infected and

recovered neighbours. This corresponds to the part of the graph that has already been revealed up to a given time. Recall the construction of the configuration model random graph where the graph is dynamically revealed as infection spreads (see Section 8.3). As a result, the hypergeometric argument as mentioned in Remark 8.3.1 seems inadequate. For the purpose of obtaining a law of large numbers, we can circumvent this difficulty by suitably bounding the jump sizes of different martingales arising in the proof by the degrees of the nodes concerned. Therefore, we actually do not need the exact neighbourhood distribution of an infected individual for deriving laws of large numbers. However, to establish an FCLT, one needs to find the limit of the quadratic covariation process that would involve the task of approximating quantities such as  $\sum_{k \in \mathbb{N}_0} \sum_{i \in I_k} X_{IS,i}^2$ , where  $I_k$  is the collection of degree- $k$  nodes that are infected and  $X_{IS,i}$  is the number of susceptible neighbours of an infected individual of degree  $k$ . We suspect an elaborate bookkeeping of the infection spreading process would be necessary to approximate such quantities. We have not been able to find a simple workaround so far and intend to pursue this problem in the near future.

In this chapter, we derived an FCLT for the simplest possible MABM. In the next chapter, we shall consider more general MABMs. We shall also adopt a different approximation strategy, namely approximate lumpability based on local symmetries of the graph.

In this chapter, we shall consider MABMs in full generality and propose approximations via approximately lumpable aggregation of states. Consider the MABM described in Section 2.5. Let  $G = (V, E)$  be a graph (possibly a realisation of a random graph), where  $V = [N] := \{1, 2, \dots, N\}$  is the set of vertices, and  $E \subseteq V \times V$  is the set of edges. For simplicity, we assume  $G$  is undirected in the sense that  $(u, v) \in E$  whenever  $(v, u) \in E$ , for  $u, v \in V$ . Let us denote the degree of vertex  $i$  by  $d_i$ . Let  $X_i(t)$  denote the local state of vertex  $i \in [N]$  at time  $t \in \mathcal{T} := [0, T]$  for some  $T > 0$ . For simplicity, we assume the vertices have the same finite local state space  $\mathcal{X}$ , i.e.,  $X_i \in \mathcal{X}$ , for all  $i \in [N]$ . We are interested in the process  $X := (X_1, X_2, \dots, X_N) \in \mathcal{X}^N$ . We assume the process  $X$  is a CTMC, whose transition rates depend on  $G$ .

In a recent paper Simon, Taylor, and Kiss (2011), the authors introduced a novel lumping procedure based on the automorphisms of the underlying graph  $G$ . They considered a stochastic Susceptible-Infected-Susceptible (SIS) epidemic process on a graph. They showed that, when the automorphism group is known, a lumpable partition can be obtained by determining the orbits of the elements of the state space with respect to the automorphism group. The idea of lumping using graph automorphisms is innovative. However, it is not always efficient for two reasons. First, finding all automorphisms without additional information about the graph structure is computationally prohibitive, especially for large graphs (see Babai (2015)). Second, there may be too few automorphisms to engender significant state space reduction (Simon, Taylor, and Kiss 2011) as many large random graphs tend to be asymmetric with high probability (see J. H. Kim, Sudakov, and Vu (2002), Łuczak (1988), and McKay and N. C. Wormald (1984)). Therefore, we propose a lumping procedure based on a *local* notion of symmetry (Elbert Simões, Figueiredo, and Barbosa 2016) taking into account only local ( $k$ -hop) neighbourhoods of each vertex. In our approach, we construct an equitable partition (Godsil and Royle 2013, Chapter 9) of  $V$  by clubbing together vertices that are locally symmetric. We say two vertices  $u$  and  $v$  are locally symmetric if there exists an isomorphism  $f$  between their respective local neighbourhoods (the induced subgraphs) such that  $f(u) = v$ . This is less restrictive than demanding the existence of an automorphism  $g$  on the entire graph  $G$  mapping  $u$  to  $v$ . We make the idea precise in the next following sections.

## 9.1 MARKOVIAN AGENT-BASED MODEL

### 9.1.1 Interaction rules and the transition intensities

The most important ingredient of an MABM are the interaction rules of the agent-based local processes  $X_i$ 's. These rules of interaction determine the dynamics of the process. Note that an MABM can also be viewed as a collection of local CTMCs that are connected to each other via the graph  $G$ . In other words, each  $X_i$  can be seen as a local CTMC, conditioned on the rest. In this work, we assume the intensities of the local CTMC  $X_i$

depend on the local states  $X_j$ 's of the neighbours of the vertex  $i \in V$  (such that  $(i, j) \in E$ ). Let  $d_i = |\{j \in V \mid (i, j) \in E\}|$  denote the number of neighbours of vertex  $i$ . Additionally, we assume the intensities depend only on the counts of neighbours for each local state  $a \in \mathcal{X}$ . Therefore, we define the following summary function  $c$  that returns population counts for different configurations of local states

$$c : \{\emptyset\} \cup \left( \bigcup_{l=1}^N \mathcal{X}^l \right) \longrightarrow \{\emptyset\} \cup \left( \bigcup_{l=1}^N \Lambda(l, K) \right)$$

such that, for  $x = (x_1, x_2, \dots, x_l) \in \mathcal{X}^l$ , and  $l \in [N]$ ,

$$c(x) = (y_1, y_2, \dots, y_K) \in \Lambda(l, K) \text{ where } y_i = |\{x_j = i \in \mathcal{X} : j = 1, 2, \dots, l\}|, \quad (9.1.1)$$

and set  $c(\emptyset) = \emptyset$ . The empty set  $\emptyset$  is used to denote the neighbourhood of an isolated vertex. An important feature of the set-valued function  $c$  is that it is permutation invariant in the sense that  $c(x) = c(x')$  if the elements of  $x'$  are permutations of the elements of  $x$ . In order to extract the neighbourhood information out of the global configuration, we define a family of set-valued functions  $n_i$  in the following way

$$n_i : \mathcal{X}^N \longrightarrow \{\emptyset\} \cup \left( \bigcup_{l=1}^{N-1} \mathcal{X}^l \right) \text{ for } i \in [N],$$

such that, for  $x = (x_1, x_2, \dots, x_N) \in \mathcal{X}^N$ ,

$$n_i(x) = \begin{cases} (x_{i_1}, x_{i_2}, \dots, x_{i_l}) & \text{if } (i, i_j) \in E \forall j = 1, 2, \dots, l \text{ and } l = d_i, \\ \emptyset & \text{otherwise.} \end{cases} \quad (9.1.2)$$

Having defined these two important functions, we now define the interaction rules by means of local transition intensities. We assume the intensities depend only on the type of local transition and the summary of the neighbourhood configuration of a vertex. Therefore, we define the local intensity function

$$\gamma : \mathcal{X} \times \mathcal{X} \times \left( \{\emptyset\} \cup \left( \bigcup_{l=1}^{N-1} \Lambda(l, K) \right) \right) \longrightarrow \mathbb{R}_+, \quad (9.1.3)$$

where we interpret  $\gamma(a, b, y)$  as the local intensity of making a transition from local state  $a$  to  $b$  by a vertex when the summary of its neighbourhood configuration is  $y$ .

We are now in a position to specify the transition rate or the infinitesimal generator matrix for our MABM  $X$ . Note that the process  $X$  jumps from a state  $x$  to  $y$  whenever one of the local processes  $X_i$ 's jumps. Therefore, only one of the coordinates of the states  $x$  and  $y$  differ. Let the  $K^N \times K^N$  matrix  $Q = ((q_{x,y}))$  denote the transition rate matrix of  $X$ . The elements of the matrix  $Q$  are given by

$$q_{x,y} = \begin{cases} \sum_{i \in [N]} \mathbb{1}(x_i \neq y_i, x_j = y_j \forall j \in V \setminus \{i\}) \gamma(x_i, y_i, c(n_i(x))) & \text{if } x \neq y, \\ -\sum_{y \neq x} q_{x,y} & \text{if } x = y. \end{cases} \quad (9.1.4)$$

We interpret  $q_{x,y}$  as the rate of transition from  $x$  to  $y$ , where  $x, y \in \mathcal{X}^N$ . For ease of understanding, we have suffixed the entries of  $Q$  by the different configurations  $x, y \in$



$\mathcal{X}^N$  and interpret them as functions on  $\mathcal{X}^N \times \mathcal{X}^N$ , instead of introducing a bijection between  $\mathcal{X}^N$  and  $[K^N]$  to label the states in a linear order so that the suffixes range over the integers from 1 to  $K^N$ . Note that the particular choice of bijection to label the states is immaterial for our purposes, because such a bijection essentially yields a permutation of  $[K^N]$ , and in the light of Proposition 2.4.3, does not alter lumpability properties of  $Q$ . Finally, we study the dynamics of  $X$  via the linear system

$$\dot{p} = pQ. \quad (9.1.5)$$

The vector-valued function  $p$  gives the probability distribution of  $X$ .

### 9.1.2 Examples

**SUSCEPTIBLE-INFECTED-SUSCEPTIBLE EPIDEMICS** The SIS epidemic model (Simon, Taylor, and Kiss 2011) captures the dynamics of an epidemic spread over a human or an animal population. It encapsulates binary dynamics in the sense that the local state space is written as  $\mathcal{X} := \{1, 2\}$ , where 1 indicates susceptibility and 2, an infected status. Infected vertices infect one of its randomly chosen neighbours at each ticking of a Poisson clock with a fixed rate  $a > 0$ . Infected vertices themselves recover to susceptibility at a rate  $b \geq 0$ , independent of the neighbours' statuses. When  $b = 0$ , the model is called a susceptible-infected (SI) model. Therefore, the local transition intensities are given by

$$\gamma(1, 2, (x_1, x_2)) = x_2 a, \quad \text{and} \quad \gamma(2, 1, (x_1, x_2)) = b.$$

We set  $\gamma$  to zero in every other case. This fully describes the dynamics of the system.

**PEER-TO-PEER LIVE MEDIA STREAMING SYSTEMS** Peer-to-peer networks are engineered networks where the vertices, called peers, communicate with each other to perform certain tasks in a distributed fashion. In particular, content delivery platforms such as BitTorrent, file sharing platforms such as Gnutella, (live) media (audio/video) streaming platforms use peer-to-peer networks. For the purposes of performance analysis, Markov chain models are often used for such systems.

In a P2P live streaming system, each peer maintains a buffer of length  $L$ . The availability of a media chunk at buffer index  $i \in [L]$  is indicated by 1, and likewise unavailability, by 0 (see KhudaBukhsh, Rückert, et al. (2016) and also Chapter 10). Therefore, local state space is given by  $\mathcal{X} = \{0, 1\}^L$ . Put  $K = 2^L$  so that  $\{0, 1\}^L$  can be put in one-to-one correspondence with  $[K]$ . The chunk at buffer index  $L$ , if available, is played back at rate unity and then removed. After playback, all other chunks are moved one index to the right, *i.e.*, the chunk at buffer index  $i$  is shifted to buffer index  $i + 1$ . The central server selects a finite number of peers at random and uploads chunks at buffer index 1. All other peers (not receiving chunks from the server) download chunks from their neighbours, following a *pull* mechanism<sup>1</sup>. The peers maintain their private Poisson clocks at the tickings of which they contact their neighbours to download missing chunks. Let the rate of these Poisson clocks be  $a > 0$ . The neighbours oblige the request if the requested chunk is available. When multiple chunks are missing, the peers prioritise the

<sup>1</sup> There are also systems where the peers *push* chunks into their neighbours' buffers instead of pulling.

chunks in some way giving rise to different chunk selection strategies, such as the Latest Deadline First (LDF) and the Earliest Deadline First (EDF) strategies. Let us introduce a function, called chunk selection function that captures this prioritisation, usually represented as probabilities. Let  $s : [L] \times \mathcal{X} \times \mathcal{X}$  be the chunk selection function. We interpret  $s(i, u, v)$  as the probability of a vertex with buffer configuration  $u$  selecting to fill buffer index  $i$  when it contacts a neighbour with buffer configuration  $v$ . Let  $y_1, y_2, \dots, y_K$  be a linear arrangement of the states in  $\mathcal{X}$ . Denote the  $j$ -th component of  $y_i$  by  $y_{i,j}$ , i.e.,  $y_i = (y_{i,1}, y_{i,2}, \dots, y_{i,L})$ . The local intensity function is then given by KhudaBukhsh, Rückert, et al. (2016)

$$\gamma(u, u + e_j, (x_1, x_2, \dots, x_K)) = a \sum_{i \in [K]} \mathbb{1}(y_{i,j} = 1) x_i s(j, u, y_i) \text{ if } j > 1,$$

where  $e_j$  is the  $j$ -th unit vector in the  $L$ -dimensional Euclidean space, and  $(x_1, x_2, \dots, x_K)$  is the population count vector of the neighbours of a vertex with different buffer configurations. Besides the above transitions due to download of a chunk from a neighbour, there are two other transitions, namely, the transition due to the shifting after playback that takes place at rate unity irrespective of the buffer configurations of the neighbours, and the transition due to being directly served by the server. The latter event also takes place irrespective of the buffer configurations of the neighbours, but a rate that depends on the exact implementation set-up of the peer-to-peer system. See KhudaBukhsh, Rückert, et al. (2016) for a detailed account on this.

## 9.2 AUTOMORPHISM-BASED LUMPING OF AN MABM

Now we discuss how graph automorphisms can be used to lump states of  $X$ . The idea was introduced by Simon, Taylor, and Kiss (2011) for SIS epidemics on graphs. The purpose of lumping states is to generate a Markov chain on a smaller state space. However, we should make sure that the loss of information is not too much. For instance,  $X$  is always lumpable with respect to the partition  $\{\mathcal{X}^N\}$ , but if all states are lumped together, all information about the dynamics of  $X$  are lost except for the fact that total probability is conserved at all times. On the other hand,  $X$  is also lumpable with respect to the partition  $\{\{x\} \mid x \in \mathcal{X}^N\}$ , which retains all the information but does not yield any state space reduction. Therefore, one needs to find a meaningful partition that yields as much state space reduction as possible with minimal loss of information. For an MABM, population counts are very useful quantities. Therefore, in order to retain information about the population counts, we first partition  $\mathcal{X}^N$  into  $\{\mathcal{X}_a \mid a \in \Lambda(N, K)\}$ , i.e.,

$$\mathcal{X}^N = \cup_{a \in \Lambda(N, K)} \mathcal{X}_a \text{ where } \mathcal{X}_a := \{b \in \mathcal{X}^N \mid c(b) = a\}, \quad (9.2.1)$$

and then seek a lumpable partition that is ideally *minimally* finer than this. The partition in (9.2.1) lumps together states that produce the same population counts. The size of this partition, i.e.,  $|\{\mathcal{X}_a \mid a \in \Lambda(N, K)\}|$ , is  $\binom{N+K-1}{K-1}$ . Note that, in the standard mean-field approach, one assumes that  $X$  is lumpable with respect to the partition in (9.2.1) and studies (approximate) CMEs corresponding to the different population counts. Next, we refine this partition using automorphisms.

A bijection  $f : V \rightarrow V$  is called an automorphism on  $G$  if  $(i, j) \in E$  if and only if  $(f(i), f(j)) \in E$ , for all  $i, j \in V$  (see Godsil and Royle (2013)). The collection of all

automorphisms forms a group under the composition of maps. This group is denoted by  $\text{Aut}(G)$ . Clearly,  $\text{Aut}(G)$  is a subgroup of  $\text{Sym}(V)$ . In order to use automorphisms to produce a partition of  $\mathcal{X}^N$ , we shall let  $\text{Aut}(G)$  act on  $\mathcal{X}^N$ . We define the following group action (a map from  $\text{Aut}(G) \times \mathcal{X}^N$  to  $\mathcal{X}^N$ )

$$f \cdot x = y \in \mathcal{X}^N \iff x_{f(i)} = y_i \forall i \in [N] \text{ for } f \in \text{Aut}(G), x \in \mathcal{X}^N. \quad (9.2.2)$$

The rationale is that, for our purpose, an automorphism needs to preserve the local states of vertices as well. Note that the action of the group  $\text{Aut}(G)$  defined above can be used to introduce an equivalence relation on  $\mathcal{X}^N$  as follows: we say  $x$  and  $y$  are equivalent with respect to the action of  $\text{Aut}(G)$ , denoted as  $x \sim y$ , if and only if there exists an  $f \in \text{Aut}(G)$  such that  $f \cdot x = y$ . The equivalence classes  $\{\tilde{\mathcal{X}}_1, \tilde{\mathcal{X}}_2, \dots, \tilde{\mathcal{X}}_M\}$  of the relation  $\sim$  yield a lumpable partition of  $\mathcal{X}^N$ . Moreover, the partition thus obtained is finer than  $\{\mathcal{X}_a \mid a \in \Lambda(N, K)\}$ . We prove this in the following.

**Proposition 9.2.1.** *The partition  $\{\tilde{\mathcal{X}}_1, \tilde{\mathcal{X}}_2, \dots, \tilde{\mathcal{X}}_M\}$  induced by the equivalence relation  $\sim$ , i.e., the quotient space  $\mathcal{X}^N / \sim$ , is a refinement of  $\{\mathcal{X}_a \mid a \in \Lambda(N, K)\}$ . That is, for each  $i \in [M]$ , there exists an  $a \in \Lambda(N, K)$  such that  $\tilde{\mathcal{X}}_i \subseteq \mathcal{X}_a$ .*

*Proof.* Pick any  $\tilde{\mathcal{X}}_i$  and  $x \in \tilde{\mathcal{X}}_i$ . Then,  $a = c(x) \in \Lambda(N, K)$ , and therefore,  $x \in \mathcal{X}_a$ . The proof completes when we show that every other  $y$  in  $\tilde{\mathcal{X}}_i$  is also in  $\mathcal{X}_a$ . Now,  $y \in \tilde{\mathcal{X}}_i$  implies  $x \sim y$ , and therefore, there exists an  $f \in \text{Aut}(G)$  such that  $f \cdot x = y$ . From the permutation invariance of  $c$ , we get  $c(y) = c(f \cdot x) = c(x) = a$  implying  $y \in \mathcal{X}_a$ .  $\square$

**Theorem 9.2.1.** *The CTMC  $X$  with transition rate matrix  $Q$  (or equivalently the linear system  $\dot{p} = pQ$ ) is lumpable with respect to the quotient space  $\mathcal{X}^N / \sim$ , the partition  $\{\tilde{\mathcal{X}}_1, \tilde{\mathcal{X}}_2, \dots, \tilde{\mathcal{X}}_M\}$  induced by the equivalence relation  $\sim$ .*

Before proving Theorem 9.2.1, we prove the following useful lemma regarding the neighbourhood function and the action of the group  $\text{Aut}(G)$ .

**Lemma 9.2.1.** *For all  $i \in [N]$  and for any  $z \in \mathcal{X}^N$ , the following is true for all  $f \in \text{Aut}(G)$ :*

$$n_{f^{-1}(i)}(f \cdot z) = n_i(z). \quad (9.2.3)$$

*Proof of Lemma 9.2.1.* Let us put  $f \cdot z = x$  and  $f^{-1}(i) = k$ . If  $d_k = 0$ , the assertion follows immediately because both sides of (9.2.3) are the empty set. Therefore, we assume  $d_k = l > 0$ . Then,

$$\begin{aligned} n_{f^{-1}(i)}(f \cdot z) &= (x_{i_1}, x_{i_2}, \dots, x_{i_l}) \text{ if } (i_j, k) \in E \forall j \in [l] \\ &= (z_{f(i_1)}, z_{f(i_2)}, \dots, z_{f(i_l)}) \text{ if } (i_j, k) \in E \forall j \in [l] \\ &= n_{f(k)}(z), \end{aligned}$$

but  $f(k) = i$  implying  $n_{f^{-1}(i)}(f \cdot z) = n_i(z)$ .  $\square$

Now we present the proof of Theorem 9.2.1.

*Proof of Theorem 9.2.1.* We check the Dynkin's criterion to establish lumpability. For any two distinct  $i, j \in [M]$ , we check if  $\tilde{q}_{i,j} = \sum_{y \in \tilde{\mathcal{X}}_j} q_{x,y} = \sum_{y \in \tilde{\mathcal{X}}_j} q_{z,y}$  for each distinct pair  $x, z \in \tilde{\mathcal{X}}_i$ . Since  $z \sim x$ , there exists an  $f \in \text{Aut}(G)$  such that  $f \cdot z = x$ . The idea is to apply  $f$  on the states of  $\tilde{\mathcal{X}}_j$  and then show that, for any two states  $x, z \in \tilde{\mathcal{X}}_i$ , there are two states  $y, f \cdot y \in \tilde{\mathcal{X}}_j$  such that the neighbourhood information are preserved.

$$\begin{aligned}
\sum_{y \in \tilde{\mathcal{X}}_j} q_{x,y} &= \sum_{y \in \tilde{\mathcal{X}}_j} \sum_{i \in [N]} \mathbb{1}(x_i \neq y_i, x_j = y_j \forall j \neq i) \gamma(x_i, y_i, c(n_i(x))) \\
&= \sum_{f \cdot y \in \tilde{\mathcal{X}}_j} \sum_{i \in [N]} \mathbb{1}(x_i \neq y_{f(i)}, x_j = y_{f(j)} \forall j \neq i) \gamma(x_i, y_{f(i)}, c(n_i(x))) \\
&= \sum_{f \cdot y \in \tilde{\mathcal{X}}_j} \sum_{i \in [N]} \mathbb{1}(z_{f(i)} \neq y_{f(i)}, z_{f(j)} = y_{f(j)} \forall j \neq i) \gamma(z_{f(i)}, y_{f(i)}, c(n_i(f \cdot z))) \\
&= \sum_{f \cdot y \in \tilde{\mathcal{X}}_j} \sum_{f^{-1}(i) \in [N]} \mathbb{1}(z_i \neq y_i, z_j = y_j \forall j \neq i) \gamma(z_i, y_i, c(n_i(z))) \\
&= \sum_{y \in \tilde{\mathcal{X}}_j} \sum_{i \in [N]} \mathbb{1}(z_i \neq y_i, z_j = y_j \forall j \neq i) \gamma(z_i, y_i, c(n_i(z))) = \sum_{y \in \tilde{\mathcal{X}}_j} q_{z,y},
\end{aligned}$$

where we have used  $n_{f^{-1}(i)}(f \cdot z) = n_i(z)$  from Lemma 9.2.1. Denoting common value by  $\tilde{q}_{i,j} = \sum_{y \in \tilde{\mathcal{X}}_j} q_{x,y}$ , the matrix  $\tilde{Q} = ((\tilde{q}_{i,j}))$  is the transition rate matrix of  $\text{agg}(X)$ .  $\square$

**Remark 9.2.1.** From the perspective of group theory, finding the lumping classes is equivalent to determining the orbits of states in  $\mathcal{X}^N$  with respect to the group  $\text{Aut}(G)$ . For a state  $x \in \mathcal{X}^N$ , the orbit of  $x$  with respect to the action of the group  $\text{Aut}(G)$ , denoted as  $\text{Aut}(G) \cdot x$ , is defined by  $\text{Aut}(G) \cdot x = \{f \cdot x \mid f \in \text{Aut}(G)\}$ .

**Example 9.2.1** (Complete graph). The automorphism group  $\text{Aut}(G)$  for the complete graph is  $\text{Sym}([N])$ . Therefore, any two states  $x, y \in \mathcal{X}^N$  can be lumped together if  $y$  is a rearrangement of components of  $x$ , i.e.,  $y = f \cdot x$  for some  $f \in \text{Sym}([N])$ . As a consequence,  $\{\mathcal{X}_a \mid a \in \Lambda(N, K)\}$  itself is a lumpable partition of  $\mathcal{X}^N$ .

**Example 9.2.2** (Star graph). An automorphism on a star graph leaves the central node (root) unchanged and permutes the rest of the nodes (leaf nodes) in any possible manner. Without loss of generality, let us assume the central node is labelled  $N$ . Then, the automorphism group  $\text{Aut}(G)$  is given by  $\text{Aut}(G) = \{g \in \text{Sym}([N]) \mid g(N) = N, g(i) = f(i) \forall i \in [N-1] \text{ for some } f \in \text{Sym}([N-1])\}$ .

**Example 9.2.3** (Cycle graph). The automorphisms of a cycle graphs are the reflections and rotations of the graph, forming a group that is also known as the dihedral group. Therefore, there are  $2N$  automorphisms. In Simon, Taylor, and Kiss (2011), the authors show that the dihedral group leads to a non-trivial lumping of states.

**Example 9.2.4** (Trees). For a star graph, we noted that an automorphism permutes the leaves but needs to leave the root unchanged. Similarly, for a tree, we start with the leaves. Any two leaves connected to the same parent node can be freely permuted.

However, whenever we permute two leaf nodes that have different parents, we also need to permute the parents to preserve the neighbourhood structure. Therefore, an automorphism on a tree necessarily maps vertices to vertices at the same height.

### 9.3 LUMPING STATES USING LOCAL SYMMETRY

In this section, we discuss lumping ideas based on a local notion of automorphism. In many cases, the number of automorphisms decrease drastically as the graph grows arbitrarily large. For instance, it is known that ER random graphs tend to be asymmetric with probability approaching unity as the size of the graph  $N$  grows to infinity (Łuczak 1988). Similar statements are true for  $d$ -regular random graphs under various sets of conditions on  $d$  relative to the number of vertices  $N$  (J. H. Kim, Sudakov, and Vu 2002), and random graphs with specified degree distributions (McKay and N. C. Wormald 1984). As a consequence, the automorphism-based lumping tends to be ineffective in state space reduction as the size of the graphs grows arbitrarily. Therefore, it is desirable to bring in a notion of local automorphism or local symmetry that would allow swapping vertices that are locally indistinguishable (*i.e.*, have similar neighbourhoods), but are not so globally. This notion of symmetry is weaker than an automorphism, which endows global symmetry on a graph. However, the potential gain is in the ability to engender state space reduction when the graph grows arbitrarily large rendering automorphism-based lumping virtually ineffective. In the following, we make these ideas precise.

#### 9.3.1 Local symmetry

There have been several attempts to formulate a more flexible notion of local symmetry. However, the literature seems divided on this and there is not a single universally accepted concept. In our set-up, it seems intuitive that two vertices that are locally indistinguishable in a large graph would also behave indistinguishably, and therefore, can be swapped. A notion of local symmetry identifying such vertices was proposed in Elbert Simões, Figueiredo, and Barbosa (2016), which we adopt in this work. We need a few definitions to make precise what we mean by two vertices being locally indistinguishable.

In order to define locality, we need some notion of distance between vertices of  $G$ . Let  $d(u, v)$  denote the smallest distance (length of the minimal path) between two vertices  $u, v \in V$ . If  $u$  and  $v$  are not connected, *i.e.*, there is no path between them, we simply set  $d(u, v) = \infty$ .

**Definition 9.3.1.** Given a vertex  $u \subseteq V$ , define its  $k$ -neighbourhood in  $G$ , denoted by  $N_k(u)$ , as follows

$$N_k : V \longrightarrow 2^V \text{ such that } N_k(u) := \{v \in V \mid d(u, v) \leq k\}. \quad (9.3.1)$$

Let  $G[N_k(u)]$  denote the subgraph of  $G$  induced by  $N_k(u)$ . The notion of locality we adopt in this work hinges on these  $k$ -neighbourhoods and their induced subgraphs. If two vertices induce isomorphic subgraphs, they are indistinguishable locally and we say they are  $k$ -locally symmetric (Elbert Simões, Figueiredo, and Barbosa 2016).

**Definition 9.3.2.** Two vertices  $u, v \in V$  are defined to be  $k$ -locally symmetric if there exists an isomorphism  $f$  between  $G[N_k(u)]$  and  $G[N_k(v)]$  such that  $f(u) = v$ .

Therefore, two vertices  $u, v \in V$  are  $k$ -locally symmetric if their  $k$ -th order local structures ( $k$ -hop neighbourhoods) are equivalent in the sense that there is a structure-preserving (edge-preserving in this case) bijection between them. When  $k = 1$ , we simply say the vertices are *locally* symmetric.

As with automorphism, local symmetries also induce an equivalence relation on the set of vertices  $V$ . We say two vertices  $u, v \in V$  are equivalent with respect to  $k$ -local symmetry, denoted by  $u \stackrel{k}{\sim} v$ , if there exists an isomorphism  $f$  between  $G[N_k(u)]$  and  $G[N_k(v)]$  such that  $f(u) = v$ . The notion of local symmetry is related to the concept of views in discrete mathematics literature (Hendrickx 2014; Yamashita and Kameda 1996). The view of depth  $k$  of a vertex is a tree containing all walks of length  $k$  leaving that vertex. However, please note that, in our context, the induced subgraphs  $G[N_k(u)]$  need not be trees. The following facts about local symmetry are useful for our study of lumpability (Elbert Simões, Figueiredo, and Barbosa 2016; Norris 1995).

**Proposition 9.3.1.** *The following properties are satisfied by  $k$ -local symmetry*

- P1* For  $u, v \in V$ ,  $u \stackrel{k+1}{\sim} v \implies u \stackrel{k}{\sim} v$ . Consequently,  $V / \stackrel{k+1}{\sim}$ , the equivalence classes of  $\stackrel{k+1}{\sim}$  form a refinement of  $V / \stackrel{k}{\sim}$ , the equivalence classes of  $\stackrel{k}{\sim}$ .
- P2* If the equivalence classes of  $\stackrel{k+1}{\sim}$  are the same as those of  $\stackrel{k}{\sim}$ , the equivalence classes of all  $\stackrel{k+j}{\sim}$  are the same as those of  $\stackrel{k}{\sim}$ , for  $j \in \mathbb{N}$ .
- P3* If  $k \geq \text{diam}(G)$ , the diameter of  $G$ , then, for two vertices  $u, v \in V$ , we have  $u \stackrel{k}{\sim} v \iff$  there exists an  $f \in \text{Aut}(G)$  such that  $f(u) = v$ . That is,  $k$ -local symmetry is equivalent to automorphism if  $k$  is as large as the diameter of  $G$ .

In addition to the above, it can be verified that the quotient spaces  $V / \stackrel{k}{\sim}$  are equitable partitions (Godsil and Royle 2013, Chapter 9) for each  $k \geq 1$ . We use these properties to lump states of  $\mathcal{X}^N$  in the next section.

### 9.3.2 Lumping states using local symmetry

The procedure to lump states in  $\mathcal{X}^N$  using local symmetry is similar to the procedure used to lump states using automorphism. However, unlike the case with automorphism, we now allow permutations that only need to ensure symmetry locally. That is, in order to lump states using  $k$ -local symmetry, we allow permuting two vertices  $u$  and  $v$  in  $V$  if and only if  $u$  and  $v$  are  $k$ -local symmetric. Therefore, define

$$\Psi_k(G) := \{f \in \text{Sym}(V) \mid f(u) = v \iff u \stackrel{k}{\sim} v, \text{ for } u, v \in V\}. \quad (9.3.2)$$

We refer to  $|\Psi_k(G)|$  as the number of local symmetries. It can be verified that  $\Psi_k(G)$ , for each  $k \geq 1$ , forms a group under the composition of maps. Therefore, we can let the group  $\Psi_k(G)$  act on  $\mathcal{X}^N$ . We define the action of  $\Psi_k(G)$  as follows

$$f \cdot x = y \in \mathcal{X}^N \iff x_{f(i)} = y_i \forall i \in [N] \text{ for } f \in \Psi_k(G), x \in \mathcal{X}^N. \quad (9.3.3)$$

Note that a state  $x$  in  $\mathcal{X}^N$  is taken to  $y$  if and only if the local states of all vertices are preserved and two vertices are swapped only when they are  $k$ -local symmetric. The above action induces the following partition of the state space: two states  $x, y \in \mathcal{X}^N$  are said to be equivalent with respect to  $k$ -local symmetry, denoted as  $x \stackrel{k}{\sim} y$ , if there exists an  $f \in \Psi_k(G)$  such that  $f \cdot x = y$ . We use the same symbol  $\stackrel{k}{\sim}$  since there is no scope of confusion. The equivalence classes of  $\stackrel{k}{\sim}$  are obtained, as before, by determining the orbits of states in  $\mathcal{X}^N$ . The orbit of a state  $x \in \mathcal{X}^N$  is given by  $\Psi_k(G) \cdot x := \{f \cdot x \in \mathcal{X}^N \mid f \in \Psi_k(G)\}$ .

The partition thus obtained (based on  $k$ -local symmetry) does not, in general, guarantee lumpability, *i.e.*,  $X$  need to be lumpable with respect to  $\mathcal{X}^N / \stackrel{k}{\sim}$ . We say  $X$  is approximately lumpable with respect to this partition and seek to quantify the approximation error in the next section. The following observation is integral to the quantification of the approximation error incurred when states of  $\mathcal{X}^N$  are lumped according to  $k$ -local symmetry instead of automorphism.

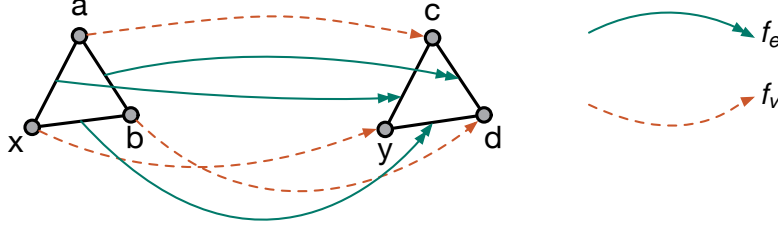
**Proposition 9.3.2.** *The quotient space  $\mathcal{X}^N / \stackrel{k+1}{\sim}$  is a refinement of  $\mathcal{X}^N / \stackrel{k}{\sim}$ .*

*Proof of Proposition 9.3.2.* Let  $\mathcal{X}_1^{(k+1)}, \mathcal{X}_2^{(k+1)}, \dots, \mathcal{X}_{M_{k+1}}^{(k+1)}$  be the equivalence classes of  $\stackrel{k+1}{\sim}$ . Also, denote the equivalence classes of  $\stackrel{k}{\sim}$  by  $\mathcal{X}_1^{(k)}, \mathcal{X}_2^{(k)}, \dots, \mathcal{X}_{M_k}^{(k)}$ . Let  $i \in [M_{k+1}]$  and  $x \in \mathcal{X}_i^{(k+1)}$ . If  $\mathcal{X}_i^{(k+1)}$  is singleton, identity map is the only map in  $\Psi_{k+1}(G)$ , but it is also in  $\Psi_k(G)$ . Therefore,  $x \in \mathcal{X}_j^{(k)}$  for some  $j \in [M_k]$ , and the assertion follows. If  $\mathcal{X}_i^{(k+1)}$  has at least two elements, say,  $x, y$ , then  $y \stackrel{k+1}{\sim} x$ . By Proposition 9.3.1, we must have  $y \stackrel{k}{\sim} x$ . Therefore, there exists a  $j \in [M_k]$  such that  $x, y \in \mathcal{X}_j^{(k)}$ . Since the choice of  $x, y$  is arbitrary, the assertion follows.  $\square$

For practical applications, one would start with  $\mathcal{X}^N / \stackrel{1}{\sim}$  and then iteratively obtain further refinements  $\mathcal{X}^N / \stackrel{2}{\sim}, \mathcal{X}^N / \stackrel{3}{\sim}$ , and so on until satisfactory accuracy is achieved (assuming we can quantify accuracy for the time being). In the light of Proposition 9.3.1, two important remarks are in place. They emphasise the benefits of local symmetry-driven lumping over the automorphism-driven one.

**Remark 9.3.1.** In an algorithmic implementation, P2 in Proposition 9.3.1 provides a stopping rule for an iterative procedure to obtain local symmetry-driven partitions. That is, we can stop at the first instance of no improvement (the equivalence classes of  $\stackrel{k+1}{\sim}$  and  $\stackrel{k}{\sim}$  are the same).





**Figure 9.1:** Fibrations map vertices to vertices and edges to edges. When three vertices form a triangle, fibrations also preserve the triangle structure. Therefore, one can define an isomorphism between local neighbourhoods using fibrations.

**Remark 9.3.2.** The diameters in many random graphs grow slowly as the number of vertices goes to infinity. For instance, the diameter of ER random graphs with  $N$  vertices and edge probability  $\lambda/N$ , for some fixed  $\lambda > 1$ , grows as  $\log N$  (Riordan and N. Wormald 2010). In the light of  $P_3$  in Proposition 9.3.1, our approach needs (at most) as many steps as the diameter of  $G$  to produce an *exactly* lumpable partition of  $\mathcal{X}^N$ . Note that  $k \geq \text{diam}(G)$  is only a sufficient condition for  $\mathcal{X}^N / \sim^k$  to be an exactly lumpable partition. For practical purposes, we may achieve sufficient accuracy (including exact lumpability) even for small values of  $k < \text{diam}(G)$ .

Our local symmetry-driven lumping approach shares a close relationship with what are known as fibrations in algebraic graph theory. We briefly describe the relationship in the following.

#### 9.4 GRAPH FIBRATIONS

Fibrations of graphs were first inspired by fibrations between a pair of categories (Boldi and Vigna 2002). Although the idea of fibrations originated from category theory, it has deep implications for graph theory, theoretical computer science, and other mathematical disciplines. For instance, in Boldi, Lonati, et al. (2006), the authors discuss its interesting connections to PageRank citation ranking algorithm. The authors in Nijholt, Rink, and Sanders (2016) explore the similarities between dynamical systems with a network structure and dynamical systems with symmetry by means of fibrations of graphs. Let us now define the necessary graph theoretic concepts.

Given the graph  $G = (V, E)$ , we first define the source and target maps  $s_G, t_G : E \rightarrow V$  on  $G$  such that  $s_G(u, v) = u$  and  $t_G(u, v) = v$  for each  $(u, v) \in E$ . Let  $H = (V', E')$  be another graph. The source and the target maps  $s_H, t_H$  are defined analogously. A map  $f := (f_v, f_e)$ , where  $f_v : V \rightarrow V'$  and  $f_e : E \rightarrow E'$ , is called a *graph morphism* between  $G$  and  $H$  (from  $G$  to  $H$ , to be precise) if  $f_v$  and  $f_e$  commute with the source and the target maps of  $G$  and  $H$ , i.e., if  $s_H f_e = f_v s_G$  and  $t_H f_e = f_v t_G$ . A morphism is called an *epimorphism* if both  $f_v$  and  $f_e$  are surjective. Finally, we define a graph fibration as follows (Boldi and Vigna 2002):



**Definition 9.4.1.** A morphism  $f := (f_v, f_e)$  between two graphs  $G = (V, E)$  and  $H = (V', E')$  is called a fibration between graphs  $G$  and  $H$  (from  $G$  to  $H$ , to be precise) if, for each edge  $a \in E'$  and for each  $x \in V$  satisfying  $f_v(x) = t_H(a)$ , there exists a unique edge  $a_x \in E$  such that  $f_e(a_x) = a$  and  $t_G(a_x) = x$ . The edge  $a_x$  thus found is called the lifting of  $a$  at  $x$ , and is denoted by  $f_e^{-1}(a)$ . The graph  $G$  is then called fibred over  $H$ . The fibre over a vertex  $y \in V'$ , denoted by  $\text{fibre}(y)$ , is the set of vertices in  $V$  that are mapped to  $y$ , i.e.,  $\text{fibre}(y) := \{x \in V \mid f_v(x) = y\}$ .

In the original paper Boldi and Vigna (2002), the authors define colour preserving graph morphisms when graphs are endowed with a colouring function. In that case,  $f_e$  also commutes with the colouring function. For our present purposes, we do not require this generality and only consider uncoloured graphs. In Boldi and Vigna (2002), the authors showed that a left action of a group on  $G$  can be used to induce fibrations. They also show that fibrations and epimorphisms satisfying certain local isomorphism property are equivalent (Boldi and Vigna 2002, Theorem 2). Indeed, fibrations have a close relationship with the notion of local symmetry described in Section 9.3. The proof of the following proposition follows analogously from Boldi and Vigna (2002, Theorem 2). However, for the sake of completeness, we also provide it in Appendix G.1.

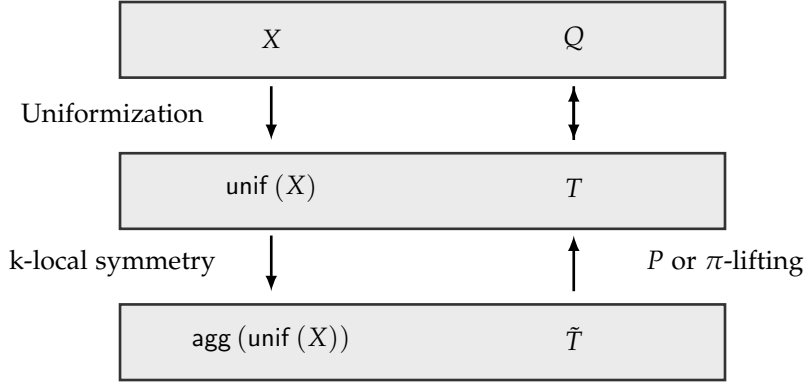
**Proposition 9.4.1.** *Let  $f := (f_v, f_e)$  be a fibration of the graph  $G = (V, E)$ , i.e., a fibration from  $G$  to  $G$  itself. Pick two vertices  $x, y \in V$ . If  $x \in \text{fibre}(y)$ , the vertices  $x, y$  are locally symmetric, i.e.,  $x \stackrel{1}{\sim} y$ . Moreover, if the vertices  $x, y$  are locally symmetric, there exists a fibration such that  $x \in \text{fibre}(y)$ .*

The above proposition essentially shows that the equivalence classes of local symmetry (with  $k = 1$ ) and fibres induced by a graph fibration are the same. Therefore, the fibres can also be used to aggregate the states of  $\mathcal{X}^N$  to achieve approximate lumpability in the same fashion as we did with local symmetry.

## 9.5 APPROXIMATION ERROR

As the lumping based on local symmetry does not ensure Markovianity of the lumped process, we need to quantify the approximation error. In order to do so, we work with the uniformization of  $X$ . Then we lump  $\text{unif}(X)$  to produce  $\text{agg}(\text{unif}(X))$  according to  $k$ -local symmetry. A direct assessment of the quality of aggregation is cumbersome. Therefore, it is suggested (Deng, Mehta, and Meyn 2011; Geiger et al. 2015) that we *lift* the aggregated process  $\text{agg}(\text{unif}(X))$  to a Markov chain on the same state space  $\mathcal{X}^N$  as  $\text{unif}(X)$  and then compare their transition probability matrices. The lifting allows us to use known metrics of divergence such as the Kullback-Leibler (KL) divergence rate to quantify the approximation error. We follow the scheme depicted in Figure 9.2.

In order to fix ideas, let us lump  $\text{unif}(X)$  according to  $k$ -local symmetry, i.e., according to the partition  $\{\mathcal{X}_1^{(k)}, \mathcal{X}_2^{(k)}, \dots, \mathcal{X}_{M_k}^{(k)}\}$  of  $\mathcal{X}^N$  obtained as the equivalence classes of  $\stackrel{k}{\sim}$ . We introduce two notations in this connection. Let  $\eta_k : \mathcal{X}^N \rightarrow [M_k]$  be the partition function associated with  $\stackrel{k}{\sim}$ , i.e.,  $\eta_k(x) := i \iff x \in \mathcal{X}_i^{(k)}$ . For  $u \in \mathcal{X}^N$ , let us denote the equivalence class containing  $u$  by  $\langle x \rangle_k$ , i.e.,  $\langle x \rangle_k := \mathcal{X}_i^{(k)} \iff x \in \mathcal{X}_i^{(k)}$ . Note that,  $\langle x \rangle_k = \eta_k^{-1}(\eta_k(x))$ .



**Figure 9.2:** Lifting procedure used to assess the quality of the approximation.

Let  $T = ((t_{ij}))$  be the transition probability matrix associated with  $\text{unif}(X)$ . Now, since  $X$  is not necessarily lumpable with respect to the partition  $\{\mathcal{X}_1^{(k)}, \mathcal{X}_2^{(k)}, \dots, \mathcal{X}_{M_k}^{(k)}\}$ , for  $i \neq j \in [M_k]$  and two distinct  $x, y \in \mathcal{X}_i^{(k)}$ , the quantity  $\sum_{z \in \mathcal{X}_j^{(k)}} t_{x,z}$  may not equal  $\sum_{z \in \mathcal{X}_j^{(k)}} t_{y,z}$ . If  $\text{unif}(X)$  is stationary with distribution  $\pi$ , i.e., if  $\pi$  is the solution to  $\pi T = \pi$  and  $p(0) = \pi$ , a natural estimate of the transition probability of the lumped process is the following

$$\tilde{t}_{i,j}^{(k)} := \frac{\sum_{u \in \mathcal{X}_i^{(k)}} \pi_u \sum_{v \in \mathcal{X}_j^{(k)}} t_{u,v}}{\sum_{u \in \mathcal{X}_i^{(k)}} \pi_u}, \text{ for } i, j \in [M_k]. \quad (9.5.1)$$

That is, we estimate the transition probabilities of the lumped process  $\text{agg}(\text{unif}(X))$  by averaging the different values  $\sum_{z \in \mathcal{X}_j^{(k)}} t_{x,z}$  and  $\sum_{z \in \mathcal{X}_j^{(k)}} t_{y,z}$ , weighted by the stationary probabilities (Geiger et al. 2015). Let  $\tilde{T}^{(k)} := ((\tilde{t}_{i,j}^{(k)}))$ . Now, we describe how the transition probabilities of the lifted Markov chain are calculated. There are two common ways of lifting  $\text{agg}(\text{unif}(X))$  to a Markov chain on  $\mathcal{X}^N$ ; one using a probability vector, called  $\pi$ -lifting, and the other using the transition probabilities, called  $P$ -lifting. Let us discuss  $\pi$ -lifting first.

**Definition 9.5.1** ( $\pi$ -lifting). The  $\pi$ -lifting of  $\eta_k(\text{unif}(X))$  is a DTMC with transition probability matrix  $T_k^\pi := ((t_{u,v}^{\pi,k}))$  where

$$t_{u,v}^{\pi,k} := \frac{\pi_v}{\sum_{x \in \langle v \rangle_k} \pi_x} \tilde{t}_{\eta_k(u), \eta_k(v)}^{(k)}, \text{ where } u, v \in \mathcal{X}^N. \quad (9.5.2)$$

Please note that, in principle,  $\pi$ -lifting can be done using any probability vector as long as the denominator remains non-zero for the choice of the candidate probability vector. Nevertheless, the most common choice is the stationary probability vector. The reason for this choice is the fact that the stationary probability vector achieves the minimum KL divergence rate (Deng, Mehta, and Meyn 2011). For this reason, we consider

$\pi$ -lifting with the stationary distribution for numerical computations in this work. Another immediate consequence of  $\pi$ -lifting is that the lifted Markov chain with transition probability matrix  $T_k^\pi$  given in (9.5.1) is lumpable with respect to the partition  $\mathcal{X}^N / \stackrel{k}{\sim}$  and has  $\pi$  as the stationary probability. Now, we define the approximation error.

**Definition 9.5.2.** We define the approximation error to be the KL divergence rate between  $\text{unif}(X)$  and the lifted DTMCs. Therefore, for  $\pi$ -lifting, the approximation error is given by

$$D_{\text{KL}}(T \parallel T_k^\pi) := \sum_{u \in \mathcal{X}^N} \sum_{v \in \mathcal{X}^N} \pi_u t_{u,v} \log \left( \frac{t_{u,v}}{t_{u,v}^{\pi,k}} \right). \quad (9.5.3)$$

Having defined the approximation error, we show that it indeed decreases monotonically with the order of local symmetry. This is precisely the assertion of Theorem 9.5.1. However, in order to prove Theorem 9.5.1, we need to make use of the following calculation, which we present in the form of a lemma.

**Lemma 9.5.1.** For any two states  $u, v \in \mathcal{X}^N$ , and for any  $k$ , define the ratio

$$\rho_k(u, v) := \frac{\sum_{t \in \langle v \rangle_k} \pi_t}{\tilde{t}_{\eta_k(u), \eta_k(v)}^{(k)}} = \frac{\sum_{p \in \langle u \rangle_k} \pi_p \sum_{q \in \langle v \rangle_k} \pi_q}{\sum_{p \in \langle u \rangle_k} \sum_{q \in \langle v \rangle_k} \pi_p t_{p,q}}. \quad (9.5.4)$$

Then, the following recursion relation holds true

$$\sum_{x \in \langle u \rangle_k} \sum_{y \in \langle v \rangle_k} \pi_x t_{x,y} \rho_{k+1}(x, y) = \rho_k(u, v) \sum_{x \in \langle u \rangle_k} \sum_{y \in \langle v \rangle_k} \pi_x t_{x,y}. \quad (9.5.5)$$

*Proof of Lemma 9.5.1.* By the refinement property of local symmetry in Proposition 9.3.2, we can find distinct integers  $i_1, i_2, \dots, i_m$  and  $j_1, j_2, \dots, j_n$  in  $[M_{k+1}]$  such that

$$\langle u \rangle_k = \cup_{l \in [m]} \mathcal{X}_{i_l}^{(k+1)} \text{ and } \langle v \rangle_k = \cup_{l \in [n]} \mathcal{X}_{j_l}^{(k+1)}. \quad (9.5.6)$$

Therefore, we can split the summation over  $\langle u \rangle_k, \langle v \rangle_k$  into disjoint equivalence classes of  $\stackrel{k+1}{\sim}$ . Within each of these equivalence classes of  $\stackrel{k+1}{\sim}$ , the quantity  $\tilde{t}_{\eta_{k+1}(x), \eta_{k+1}(y)}$  is constant, and therefore, can be pulled out of the summation. Therefore,

$$\begin{aligned} & \sum_{x \in \langle u \rangle_k} \sum_{y \in \langle v \rangle_k} \pi_x t_{x,y} \rho_{k+1}(x, y) \\ &= \sum_{p \in [m]} \sum_{q \in [n]} \sum_{x \in \mathcal{X}_{i_p}^{(k+1)}} \sum_{y \in \mathcal{X}_{j_q}^{(k+1)}} \pi_x t_{x,y} \left( \frac{\sum_{s \in \langle x \rangle_{k+1}} \pi_s \sum_{t \in \langle y \rangle_{k+1}} \pi_t}{\sum_{s \in \langle x \rangle_{k+1}} \sum_{t \in \langle y \rangle_{k+1}} \pi_s t_{s,t}} \right) \\ &= \sum_{p \in [m]} \sum_{q \in [n]} \left( \frac{\sum_{s \in \mathcal{X}_{i_p}^{(k+1)}} \pi_s \sum_{t \in \mathcal{X}_{j_q}^{(k+1)}} \pi_t}{\sum_{s \in \mathcal{X}_{i_p}^{(k+1)}} \sum_{t \in \mathcal{X}_{j_q}^{(k+1)}} \pi_s t_{s,t}} \right) \sum_{x \in \mathcal{X}_{i_p}^{(k+1)}} \sum_{y \in \mathcal{X}_{j_q}^{(k+1)}} \pi_x t_{x,y} \\ &= \sum_{x \in \langle u \rangle_k} \sum_{y \in \langle v \rangle_k} \pi_x \pi_y = \rho_k(u, v) \sum_{x \in \langle u \rangle_k} \sum_{y \in \langle v \rangle_k} \pi_x t_{x,y}. \end{aligned}$$

This completes the proof.  $\square$

Note that  $\rho_k(u, v) = \rho_k(x, y)$  for any  $u \stackrel{k}{\sim} x$  and  $v \stackrel{k}{\sim} y$ . Therefore, we can use the shorthand notation  $\rho_k(\mathcal{X}_i^{(k)}, \mathcal{X}_j^{(k)})$  to mean  $\rho_k(u, v)$  for any  $u \in \mathcal{X}_i^{(k)}, v \in \mathcal{X}_j^{(k)}$ .

**Remark 9.5.1** (Averaging argument). The main implication of Lemma 9.5.1 is that the quantity  $\rho_k(u, v)$  can be seen as a weighted average of  $\rho_{k+1}(x, y)$  where  $x, y$ 's are in the equivalence classes of  $\stackrel{k+1}{\sim}$ . The weights are precisely

$$W_{\langle u \rangle_k, \langle v \rangle_k}(\mathcal{X}_{i_p}^{(k+1)}, \mathcal{X}_{j_q}^{(k+1)}) := \frac{\sum_{x \in \mathcal{X}_{i_p}^{(k+1)}} \sum_{y \in \mathcal{X}_{j_q}^{(k+1)}} \pi_x t_{x,y}}{\sum_{x \in \langle u \rangle_k} \sum_{y \in \langle v \rangle_k} \pi_x t_{x,y}}, \quad (9.5.7)$$

where we have partitioned  $\langle u \rangle_k$  and  $\langle v \rangle_k$  into  $\mathcal{X}_{i_p}^{(k+1)}$ 's and  $\mathcal{X}_{j_q}^{(k+1)}$ 's respectively as shown in (9.5.6). We interpret  $W_{\langle u \rangle_k, \langle v \rangle_k}(\mathcal{X}_{i_p}^{(k+1)}, \mathcal{X}_{j_q}^{(k+1)})$  as the weight for the cross-section  $\mathcal{X}_{i_p}^{(k+1)} \times \mathcal{X}_{j_q}^{(k+1)}$  with regards to the partition of  $\langle u \rangle_k$  and  $\langle v \rangle_k$  given in (9.5.6). Therefore, it follows from Lemma 9.5.1 that

$$\rho_k(\langle u \rangle_k, \langle v \rangle_k) = \sum_{p \in [m]} \sum_{q \in [n]} \rho_{k+1}(\mathcal{X}_{i_p}^{(k+1)}, \mathcal{X}_{j_q}^{(k+1)}) W_{\langle u \rangle_k, \langle v \rangle_k}(\mathcal{X}_{i_p}^{(k+1)}, \mathcal{X}_{j_q}^{(k+1)}). \quad (9.5.8)$$

Since the weights sum up to unity,  $\rho_k(u, v)$  can be indeed seen as an average. Keeping this remark in mind, we now proceed to state and prove Theorem 9.5.1 about the monotonicity of the approximation error.

**Theorem 9.5.1.** *For  $\pi$ -lifting, the aggregation of states in  $\mathcal{X}^N$  using local symmetry ensures monotonically decreasing approximation error with increasing order of local symmetry. That is,*

$$D_{\text{KL}}(T \parallel T_{k+1}^{(\pi)}) \leq D_{\text{KL}}(T \parallel T_k^{(\pi)}) \text{ for all } k \geq 1. \quad (9.5.9)$$

*Proof of Theorem 9.5.1.* By the refinement property of local symmetry proved in Proposition 9.3.2, we can partition  $[M_{k+1}] = \{1, 2, \dots, M_{k+1}\}$  into  $\{\Lambda_1, \Lambda_2, \dots, \Lambda_{M_k}\}$  such that

$$\mathcal{X}_i^{(k)} = \cup_{l \in \Lambda_i} \mathcal{X}_l^{k+1}.$$

Note that

$$\begin{aligned} & D_{\text{KL}}(T \parallel T_k^{(\pi)}) - D_{\text{KL}}(T \parallel T_{k+1}^{(\pi)}) \\ &= \sum_{i,j \in [M_k]} \sum_{u \in \mathcal{X}_i^{(k)}} \sum_{v \in \mathcal{X}_j^{(k)}} \pi_u t_{u,v} \log \left( \frac{\rho_k(u, v)}{\rho_{k+1}(u, v)} \right) \\ &= \sum_{i,j \in [M_k]} (\log(\rho_k(\mathcal{X}_i^{(k)}, \mathcal{X}_j^{(k)}))) \sum_{u \in \mathcal{X}_i^{(k)}} \sum_{v \in \mathcal{X}_j^{(k)}} \pi_u t_{u,v} - \sum_{u \in \mathcal{X}_i^{(k)}} \sum_{v \in \mathcal{X}_j^{(k)}} \pi_u t_{u,v} \log(\rho_{k+1}(u, v)) \\ &= \sum_{i,j \in [M_k]} \Theta_{i,j}, \end{aligned}$$

where

$$\begin{aligned}
\Theta_{i,j} &:= \log \left( \rho_k(\mathcal{X}_i^{(k)}, \mathcal{X}_j^{(k)}) \right) \sum_{u \in \mathcal{X}_i^{(k)}} \sum_{v \in \mathcal{X}_j^{(k)}} \pi_u t_{u,v} \\
&\quad - \sum_{p \in \Lambda_i} \sum_{q \in \Lambda_j} \sum_{u \in \mathcal{X}_p^{(k+1)}} \sum_{v \in \mathcal{X}_q^{(k+1)}} \pi_u t_{u,v} \log \left( \rho_{k+1}(u, v) \right) \\
&= \left( \sum_{u \in \mathcal{X}_i^{(k)}} \sum_{v \in \mathcal{X}_j^{(k)}} \pi_u t_{u,v} \right) \times \left( \log \left( \rho_k(\mathcal{X}_i^{(k)}, \mathcal{X}_j^{(k)}) \right) \right. \\
&\quad \left. - \sum_{p \in \Lambda_i} \sum_{q \in \Lambda_j} W_{\mathcal{X}_i^{(k)}, \mathcal{X}_j^{(k)}}(\mathcal{X}_p^{(k+1)}, \mathcal{X}_q^{(k+1)}) \log \left( \rho_{k+1}(\mathcal{X}_p^{(k+1)}, \mathcal{X}_q^{(k+1)}) \right) \right) \\
&\geq 0,
\end{aligned}$$

by Jensen's inequality and the averaging argument given in Remark 9.5.1 and Lemma 9.5.1. This completes the proof.  $\square$

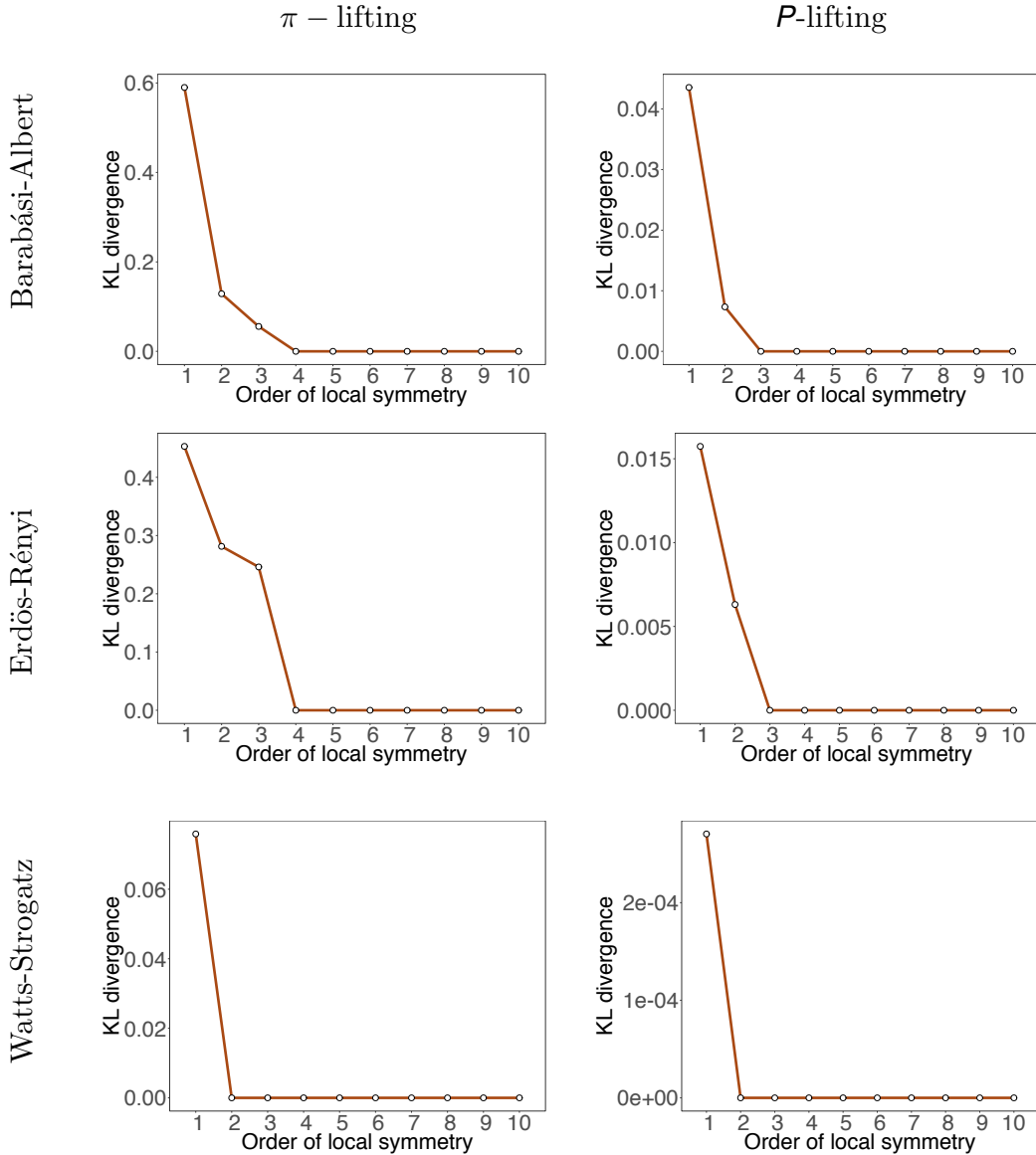
Note that  $D_{\text{KL}}(T \parallel T_k^{(\pi)}) - D_{\text{KL}}(T \parallel T_{k+1}^{(\pi)}) = 0$  is achieved if (and only if) equality is achieved in Jensen's inequality forcing the individual  $\Theta_{i,j}$ 's to be zeros. This is the case when the  $\rho_k$  and  $\rho_{k+1}$ 's are equal. There are two possibilities. First, the equivalence classes of  $\sim^k$  and  $\sim^{k+1}$  are the same. In this case, by Proposition 9.3.1, the equivalence classes of all  $\sim^{k+j}$ , for  $j \geq 2$ , will remain the same. Therefore, we have already reached automorphism, and hence, exact lumpability. Second, the equivalence classes of  $\sim^k$  and  $\sim^{k+1}$  are different (so, we are not yet at automorphism), but exact lumpability has already been achieved at order of local symmetry  $k$ . In both cases, we need not refine our partition further because exact lumpability has been achieved. Therefore,  $D_{\text{KL}}(T \parallel T_k^{(\pi)}) - D_{\text{KL}}(T \parallel T_{k+1}^{(\pi)}) = 0$  serves as a definite stopping rule for any iterative algorithmic implementation of local symmetry-driven lumping.

Now, we discuss the second type of lifting, which makes of the transition probabilities and is called  $P$ -lifting. The following is the definition.

**Definition 9.5.3** ( $P$ -lifting). The  $P$ -lifting of  $\eta_k(\text{unif}(X))$  is a DTMC with transition probability matrix  $T_k^P := ((t_{u,v}^{P,k}))$  where, for  $u, v \in \mathcal{X}^N$ ,

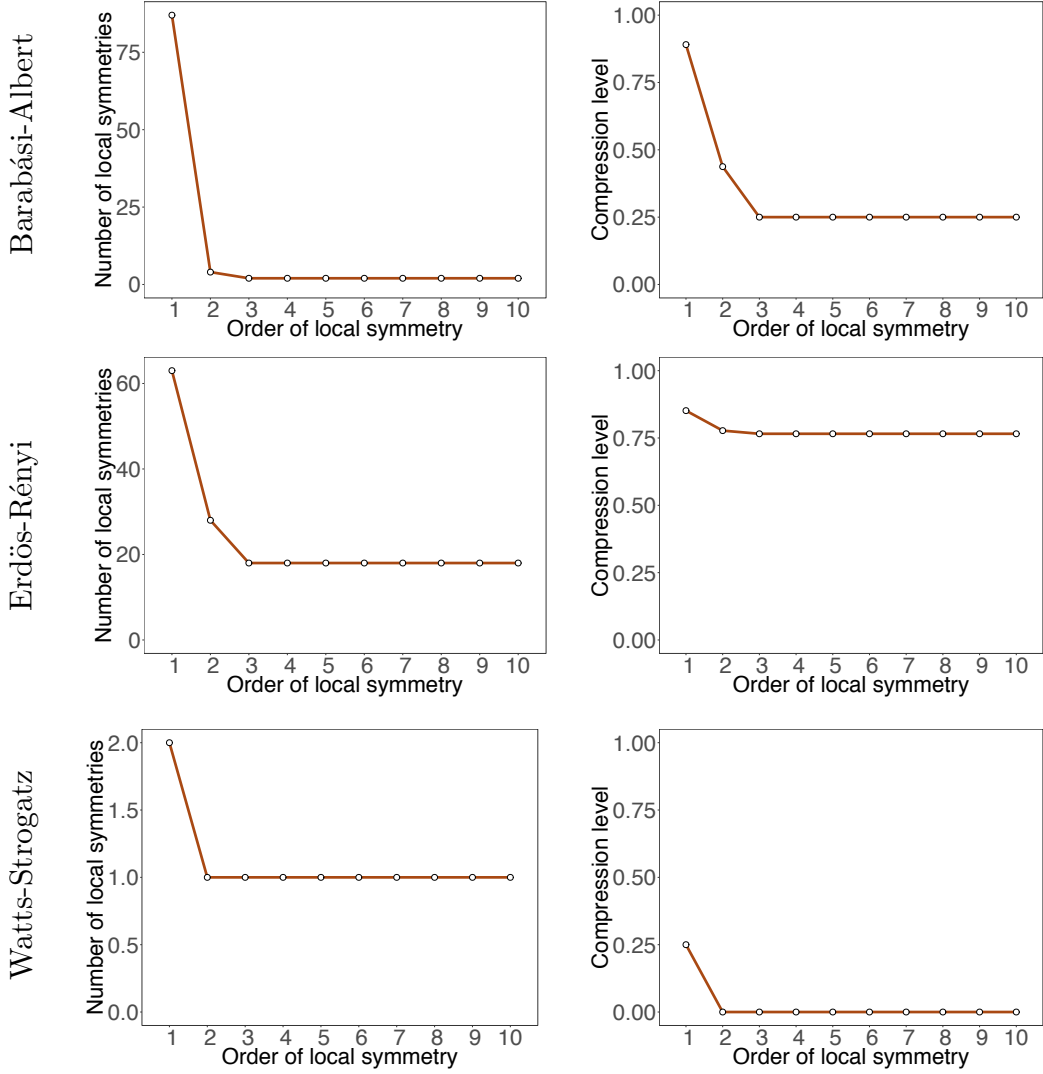
$$t_{u,v}^{P,k} := \begin{cases} \frac{t_{u,v}}{\sum_{x \in \langle v \rangle_k} t_{u,x}} \tilde{t}_{\eta_k(u), \eta_k(v)}^{(k)} & \text{if } \sum_{x \in \langle v \rangle_k} t_{u,x} > 0, \\ \frac{1}{|\langle v \rangle_k|} \tilde{t}_{\eta_k(u), \eta_k(v)}^{(k)} & \text{if } \sum_{x \in \langle v \rangle_k} t_{u,x} = 0. \end{cases} \quad (9.5.10)$$

The approximation error for  $P$ -lifting is defined similarly. Note that  $P$ -lifting is sharp, in the sense that if the lumping is in fact exact, then  $D_{\text{KL}}(T \parallel T_k^{(P)}) = 0$ , whereas  $\pi$ -lifting is not (Geiger et al. 2015). In Figure 9.3, we show numerical results pertaining to Theorem 9.5.1. We consider the Barabási-Albert (BA) preferential attachment, the ER and the WS small-world random graphs. The claimed monotonicity is observed in all three cases. In fact, the KL divergence rate steeply decreases in all three cases, for both  $\pi$  as well as  $P$ -lifting. The figures are particularly encouraging in that satisfactory level of accuracy is achieved even for small orders of local symmetry. Since one of



**Figure 9.3:** Monotonicity of the KL divergence with the order of the local symmetry for the SIS dynamics on different models of random graphs with 10 vertices. All graphs are undirected. The ER graphs are created with edge probability 0.3, while the Watts-Strogatz (WS) small world networks are created with rewiring probability 0.3 and each vertex having three neighbours. The infection and the recovery rates are both 0.5.

the main purposes of aggregation is to engender state space reduction, we need to evaluate the performance of local symmetry-driven aggregation in terms of some notion



**Figure 9.4:** As we increase the order, the number of local symmetries (the cardinality of  $\Psi_k$ ) decreases drastically. Therefore, the compression level also decreases. The simulation set-up is the same as in Figure 9.3.

of compression level as well. Therefore, we define compression level  $C$  at order of local symmetry  $k$  as follows:

$$C(k) = 1 - \frac{M_k}{|\mathcal{X}^N|}, \quad (9.5.11)$$

where  $M_k$  is the cardinality of the quotient space  $\mathcal{X}^N / \sim^k$ , i.e., the number of equivalence classes of  $\sim^k$ . If there is no non-trivial local symmetries, the compression level is zero because the partition is simply  $\{\{x\} \mid x \in \mathcal{X}^N\}$ . In Figure 9.4, we show how the number

of local symmetries decreases drastically as we increase the order of local symmetry. Consequently, the compression level also falls steeply. This is expected because random graphs tend to become asymmetric as the number of vertices increases.

**Remark 9.5.2.** Please note that Theorem 9.5.1 holds true for a general Markov chain whenever the partition function  $\eta_{k+1}$  is a refinement of  $\eta_k$ . The fact that the partition functions  $\eta_k, \eta_{k+1}$  are associated with the equivalence relations generated by  $k$  and  $k+1$ -local symmetries is only sufficient for the validity of Theorem 9.5.1, but not necessary. In fact, similar monotonicity can be proved, in similar fashion, even when  $\eta_k, \eta_{k+1}$  are arbitrary partition functions defined on the state-space of a Markov chain such that  $\eta_{k+1}$  is a refinement of  $\eta_k$ . Notably, such monotonicity can only be guaranteed for  $\pi$ -lifting. In Appendix G.1, we provide a counterexample to establish that such monotonicity fails for  $P$ -lifting when arbitrary partition functions (one being a refinement of the other) are considered. However, this observation is about a general Markov chain. For our MABM, we observe similar monotonicity for  $P$ -lifting using numerical computations, as shown in Figure 9.3, but we can not guarantee monotonicity in general.

## 9.6 DISCUSSIONS

The idea of using Markov chain lumpability for model reduction has been discussed in the literature for some years now. For instance, the authors in Kiss, Miller, and Simon (2017), Simon and Kiss (2012), and Simon, Taylor, and Kiss (2011) considered epidemiological scenarios, focussing mainly on binary dynamics. More general Markovian agent-based models were considered in Banisch (2016). Lumpability abstractions were applied to rule-based systems in Feret et al. (2012) from a theoretical computer science perspective. While model reduction is one of the main purposes of lumpability, it is not the only one. In a recent paper Katehakis and Smit (2012), the authors identify a class of Markov chains, which they call successively lumpable and for which the stationary probabilities can be computed successively by computing stationary probabilities of a cleverly constructed sequence of Markov chains (typically on much smaller state spaces).

**COVERINGS AND COLOUR REFINEMENTS** For undirected graphs, a notion similar to our notion of local symmetry is called a *covering* (Angluin 1980). However, in general, finding coverings is computationally challenging (Kratochvíl, Proskurowski, and Telle 1998). In our formulation, undirected graphs are to be treated as directed graphs with an edge set  $E$  satisfying  $(i, j) \in E \iff (j, i) \in E$ . The second notion that is similar to our approach is that of colour refinement (Arvind et al. 2016; Berkholz, Bonsma, and Grohe 2013). In order to draw analogy, we think of the local states as colours, i.e., we have a  $K$ -colouring of  $G$ , and require isomorphisms to be colour-preserving. The objective is to devise a colouring method (given the initial colouring) that is *stable* in that two vertices with the same colour have identically coloured neighbourhoods. Note that a colouring naturally induces an equivalence relation on  $V$ . With successive refinement of colouring, we can construct equitable partitions of  $V$  in much the same



way we did with local symmetry. The equitable partitions, in turn, can be used to yield approximately lumpable partitions of  $\mathcal{X}^N$ . Colour refinements are convenient and are often used as a simple isomorphism check. However, a limitation of this approach is that colour refinements are insufficient to find local isomorphisms in certain graphs such as regular graphs. In general, a graph  $G$  is said to be amenable to colour refinement if it is distinguishable from any other graph  $H$  via colour refinement. A number of classes of graphs are known to be amenable (Arvind et al. 2016), *e.g.*, unigraphs, trees. It is also known (Babai, Erdős, and Selkow 1980) that ER random graphs are amenable with high probability.

**REGULAR GRAPHS** Large regular graphs, in general, can exhibit different dynamics on them. Since the vertices have similar neighbourhoods, our local symmetry will not be able to distinguish between them. This may lead to poor lumping. Increasing the order of local symmetry will avoid such issues. A theoretical analysis of this special case of regular graphs is planned for future work.

**COMPUTATION OF THE STATIONARY DISTRIBUTION** Note that computation of the stationary distribution itself is cumbersome for Markov chains on large state spaces. In many cases, the transition matrix is sparse, which makes available a host of numerical tools developed for sparse matrices. There are also numerical algorithms (Stewart 2000), such as the Courtois’ method (Courtois 2014) or the Takahashi’s iterative aggregation-disaggregation method (Takahashi 1975), for computing the stationary distribution. In general, the efficiency of the Takahashi’s algorithm depends on a good initial clustering of states. In our case, the computation is facilitated by the fact that the initial quotient space  $\mathcal{X}^N / \sim^1$  is expectedly a better partition than a random one. In a recent paper Kuntz et al. (2017), the authors derive bounds on the stationary distribution (and moments) based on mathematical programming. In particular, when the stationary distribution is unique, they provide computable error bounds. Sampling-based techniques can also be used for this purpose. For instance, in Hemberg and Barahona (2008), the authors provide an algorithm that combine Gillespie’s algorithm with the Dominated Coupling From The Past (DCFTP) techniques to provide guaranteed sampling from the stationary distribution.

**MARKOV CHAIN ENLARGEMENT** An interesting concept closely related to aggregation is Markov chain enlargement. There are many examples where enlargement of the state space of a Markov chain can be computationally beneficial in that it can significantly reduce the mixing time. See Apers, Ticozzi, and Sarlette (2017) and F. Chen, Lovász, and Pak (1999) for a discussion on how splitting up states of a Markov chain can speed up mixing. This has implications for the performance of statistical inference algorithms that rely on the mixing of some Markov chain, and also for optimisation algorithms such as the Alternating Direction Method of Multipliers (ADMM). In França and Bento (2017), the authors show that, for certain objective functions, the distributed ADMM algorithm can indeed be seen as a lifting of the gradient descent algorithm.

**CTBNS AND SANS** The Markovian agent-based model that we consider in this work belongs to a more general class of models known as IPSs in the probability literature.

The tools developed in this work are expected to find applications beyond what has been discussed here and are immediately applicable to many of the traditional IPS models arising from statistical physics, population biology and social sciences. Such models include contact processes, voter models, exclusion models. The MABM model discussed in the present work is also closely related to Continuous Time Bayesian Networks (CTBNs) (Nodelman, Shelton, and Koller 2002) and Stochastic Automata Networks (SANs) (Buchholz and Kemper 2004). To be specific, the local intensity functions defined in (9.1.3) constitute the Conditional Intensity Matrix (CIM) in Nodelman, Shelton, and Koller (2002). These CIMs can be then combined into  $Q$  by the “amalgamation” operation. Another approach that is popular in SAN literature is to give  $Q$  a Kronecker representation (Buchholz and Kemper 2004). We expect the present endeavour will benefit and bridge the gap between the different communities that make use of the ABMs.

In this chapter, we devised a local symmetry-driven lumping procedure for MABMs. We discussed the connections between the problem of finding a (approximately) lumpable partition of the configuration space and various graph theoretic concepts such as the fibrations of the underlying graph. In the next chapter, we shall consider a P2P live media streaming scenario as an application area and construct a mixed chunk selection strategy called SCHEDMIX based on a mean-field theoretic approximations of buffer probabilities.

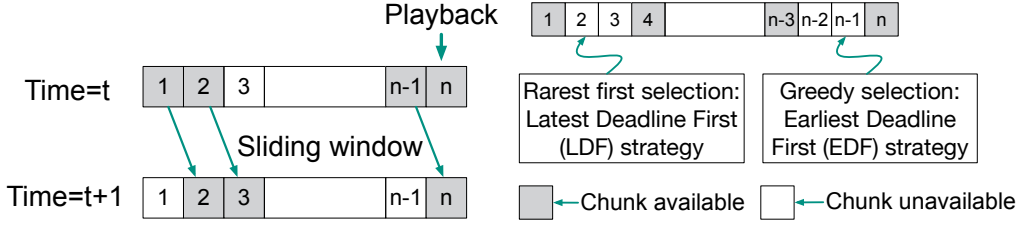
We consider a P2P live media streaming scenario as an application area in this chapter. Over the years, different classes of P2P live streaming approaches were proposed (X. Zhang and Hassanein 2012), such as tree/push- and mesh/pull-based, as well as hybrid approaches. Due to their inherent robustness, mesh/swarming approaches continue to be of major importance, especially in hybrid settings where they often function as a substrate even when tree structures run on top of them (Rückert et al. 2015; Wang, Xiong, and J. Liu 2010). A key design issue in swarming is the data scheduling strategy used by individual peers to select chunks to be requested from their neighbours. Not only must it ensure continuous playback for an individual client, but also a healthy data replication to avoid content bottlenecks (Rejaie and Magharei 2014).

Several scheduling strategies of varying levels of complexity were proposed in the literature (X. Zhang and Hassanein 2012). The impact of resource heterogeneity as observed in real client populations, however, is not yet fully understood. This leaves a big gap in the design space of practical P2P streaming approaches, where systematically leveraging resource imbalances could help in simplifying complex scheduling strategies or designing new ones. We contribute to closing this gap by analysing the basic scheduling strategies Earliest Deadline First (EDF) and Latest Deadline First (LDF) based on a mean-field theoretic analysis of a large MABM. Driven by the resulting analytic insights and with a view to designing distribution strategies for the networking scenario described in Section 1.1, we combine EDF and LDF into a simple, yet powerful, mixed strategy called SCHEDMIX to leverage differences in upload resources. SCHEDMIX assigns LDF to a small percentage of strong peers in the system and allows the majority to play greedy (EDF). The idea is to let a small percentage of strong peers act as *pseudo-servers* and facilitate propagation of new chunks. We also justify why the strong peers should agree to play LDF from a game theoretic perspective.

## 10.1 MODEL

### 10.1.1 The approach

First, we briefly explain our modelling strategy. The main idea is to model a swarming-based P2P live streaming system as a contact process on a random graph, where the vertices represent the peers. We endow each peer with a buffer of length  $n$  (a vector of 0's and 1's with 1's representing the availability of chunks). The different possible buffer configurations constitute the *local* states of a peer, which changes over time as the peer downloads chunks (from the server or from one of its neighbours following a chunk selection strategy, such as EDF or LDF), or deletes chunks that are already played back. The interactions among the peers thus define a contact process. The matrix with as many rows as the number of peers in the system and whose  $i$ -th row is the buffer configuration of the  $i$ -th peer captures the *global* state of the entire system. Our goal is to



**Figure 10.1:** (Left) The buffer as a sliding window. (Right) The two extremes: greedy selection and rarest first selection.

choose a chunk selection strategy that maximises the probability of the system being in a state that ensures good playback performance, *e.g.*, a state in which the current chunk required for playback is available at every buffer (to ensure playback continuity). In particular, the buffer probabilities (of chunk availability) can be expressed as functions of the chunk selection strategy, and therefore, can be utilised to improve the chunk selection strategy or devise a new one. This is precisely our plan.

#### 10.1.2 The network

We assume the underlying network is a realisation of a random graph. We shall be working with large graphs. We assume the associated degree distribution has a finite mean. Let  $\mathcal{G}_M$  be the class of all simple and connected random graphs with  $M$  nodes. Let  $d_l$  denote the degree of vertex  $l$ . Let  $\pi : \mathbb{N} \rightarrow [0, 1]$  be the associated degree distribution. Define the size-biased degree distribution,  $q$  as follows

$$q(k) := \frac{k\pi(k)}{\sum_k k\pi(k)}, \quad (10.1.1)$$

for  $k \in \mathbb{N}$ . The quantity  $q(k)$  is the probability that a given edge points to a vertex of degree  $k$ . The distribution  $q$  is needed to approximate the neighbourhoods of the peers.

#### 10.1.3 The peer-to-peer communication system

Suppose there are  $M$  peers and a single server. Let  $n$  denote the buffer length. The server uniformly selects a peer at random and uploads a chunk at buffer position 1. It continues to upload chunks to the chosen peer until there is a connection breakage/loss (an event that occurs with a small probability, say  $\varepsilon \in (0, 1]$ ) in which case the server again chooses a peer uniformly at random. The chunk at buffer position  $n$ , if available, is pushed for playback. After playback, the chunk is removed and all other chunks are shifted one index closer to playback (see Figure 10.1). Each peer maintains a Poisson clock with rate proportional to its degree<sup>1</sup>. A peer, if not selected by the server, contacts one of its neighbours uniformly at random at each tick of its Poisson clock and seeks to download a missing chunk. The chunk it downloads from among all downloadable

<sup>1</sup> That is, we place a Poisson clock on each edge of the graph.

chunks is decided by its chunk selection strategy. For simplicity, we assume that the playback rate is one chunk per unit of time.

Now, we describe the dynamics. The idea is to start with the exact description of the process and then gradually approximate it maintaining tractability. The approximation is carried out in two steps. First, we show that the state space grows unmanageably large, and hence we reduce the state space by means of Markov chain aggregation (see (Kemeny, Snell, et al. 1960) and also Chapter 9). Second, we perform a mean-field theoretic analysis on the aggregated chain.

#### 10.1.4 Exact description

Let  $G := (V, E) \in \mathcal{G}_M$  be a given realisation of a random graph, where  $V$  and  $E \subseteq V \times V$  are the sets of vertices and edges, respectively. Each node is a peer. Let  $\Omega := \{\omega \in \{0, 1\}^{M \times n} \mid \sum_{i=1}^M \omega(i, 1) = 1\}$  be the configuration space of all peers and buffers<sup>2</sup>, and denote all subsets of  $\Omega$  by  $2^\Omega$ . Define a CTMC  $\{X_t\}_{t \geq 0}$  on the measurable space  $(\Omega, 2^\Omega)$  as  $X_t(i, j) := 1$  if the  $j$ -th buffer location of the  $i$ -th peer is filled, and 0 otherwise. The rows of the matrix  $X_t$ , denoted as  $X_t^1, X_t^2, \dots, X_t^M$  represent buffer states of peers  $1, 2, \dots, M$ .

Let  $\mathbb{S} : \{0, 1\}^{M \times n} \cup \{0, 1\}^n \rightarrow \{0, 1\}^{M \times n} \cup \{0, 1\}^n$  denote the buffer shifting operator defined as  $\mathbb{S}Y := (0, y_1, y_2, \dots, y_{n-1})$  for  $Y = (y_1, y_2, \dots, y_n) \in \{0, 1\}^{M \times n} \cup \{0, 1\}^n$  where  $y_1, y_2, \dots, y_n$  denote the columns of  $Y$ . This operator is required to denote the state transition after a chunk, if available, is played back (see Figure 10.1 for the sliding window representation of a buffer). Let us now define the transition rates of interaction for a node  $v \in V$  as follows

$$\mu^v(u, u + e_i) = \begin{cases} \sum_{l \in V: (v, l) \in E} \varsigma \mathbb{1}(X_t(l, i) = 1) \alpha^v(i, u, X_t^l) & \text{if } i \neq 1, \\ \mathbb{1}(X_t(v, 1) = 1)(1 - \varepsilon + \varepsilon/M) + \mathbb{1}(X_t(v, 1) = 0)\varepsilon/M & \text{if } i = 1, \end{cases} \quad (10.1.2)$$

where  $u = (u_1, u_2, \dots, u_n) \in \mathcal{X} := \{0, 1\}^n$ ,  $i \in [n]$ , such that  $u_i = 0$ ,  $\varsigma > 0$  is a constant,  $e_i$  is the  $i$ -th unit basis vector of the  $n$ -dimensional Euclidean space and  $\alpha^v : [n] \times \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$  is the chunk selection function of the peer  $v \in V$ . In words,  $\alpha^v(i, u, X_t^l)\delta t$  is the probability of downloading chunk  $i$  when peer  $v$  is in buffer state  $u$  and contacts peer  $l$  in buffer state  $X_t^l$ . We defer an elaborate discussion of the chunk selection function to a later section. The case  $i \neq 1$  captures the state transition of peer  $v$  due to the successful download of a chunk at buffer location  $i$  from one of the neighbours of peer  $v$ . The case  $i = 1$  considers state transition as a result of direct upload by the server at buffer index 1. The two terms corresponding to the case  $i = 1$  differentiate whether the peer  $v$  was already connected (or reconnected after a link breakage that takes place with probability  $\varepsilon$ ) to the server and continued to receive chunks, or it was newly connected

<sup>2</sup> The server can upload a chunk at buffer index 1 to only one peer. Therefore, all the entries of the first column of the matrix are zeros except for one that corresponds to the peer receiving a chunk directly from the server.

to the server after a link breakage (between the server and some other peer). The system is described by the following CME

$$\begin{aligned} \frac{d}{dt}P(X) = & -P(X) + \sum_{v' \in V} \mathbb{1}(X(v', 1) = 1) \left[ \sum_{Y: \mathbb{S}Y = X - \Delta(v', 1)} \mu^{v'}(Y^{v'}, Y^{v'} + e_1) \left\{ P(Y) \right. \right. \\ & \left. \left. + \sum_{i \in [n] \setminus \{1\}} \sum_{v \in V \setminus \{v'\}} \left( \sum_{Z: Z = Y - \Delta(v, i)} \mu^v(Y^v - e_i, Y^v) P(Z) - \mu^v(Y^v, Y^v + e_i) P(Y) \right) \right\} \right], \end{aligned} \quad (10.1.3)$$

for  $X \in \Omega$ , where  $\Delta(v, i)$  is an  $M \times n$  matrix of all zeroes except for a unity at position  $(v, i)$ . We have used a short-hand notation  $P(X)$  to denote  $P(X_t = X)$ . The terms on right hand side correspond to the influx and outflux of probabilities of observing a particular configuration. Terms that are subtracted denote the outflux to configurations that are reachable from the current state after either a shifting of buffers or some peer downloading a chunk. On the other hand, terms that carry positive coefficients denote influx to configurations that can reach the current state after either shifting of buffers or some peer downloading a chunk. Since shifting is assumed to take place at rate unity, terms corresponding to shifting have coefficient unity, while others have their respective rates as coefficients.

#### 10.1.5 Aggregation

The CME (10.1.3) can not be solved analytically. We, therefore, carry out an aggregation of the chain into population counts. Define  $H_G := \{d \mid \exists v \in V, d_v = d\}$ . The set  $H_G$  is the set of distinct degrees realised in  $G$ . Consider a map  $A$  defined by  $A(X) := (z_x^k : x \in \mathcal{X}, k \in H_G)$  where  $z_x^k := \sum_{v \in V} \mathbb{1}(X^v = x) \mathbb{1}(d_v = k)$ , the number of degree- $k$  peers at buffer configuration  $x$ . Define an equivalence relation  $\overset{A}{\sim}$  on  $\Omega$  as  $X \overset{A}{\sim} Y \iff A(X) = A(Y)$  and  $\Omega_a := \{X \in \Omega : A(X) = a\}$  for each  $a$ . Then,  $\{\Omega_a\}$  is a partition of  $\Omega$  and each  $\Omega_a$  is an equivalence class. The induced probability is given by

$$P(A(X_t) = a) = \sum_{X \in \Omega: A(X) = a} P(X_t = X). \quad (10.1.4)$$

Such an aggregation is useful in reducing the state space if we now consider the lumped process  $A(X_t)$  of population counts instead. In Appendix H.1, we provide a necessary and sufficient condition for such an aggregation to engender state space reduction and also discuss worst case scenarios. We emphasise that we do lose information in the process of aggregation. Also, the lumped process is not necessarily Markovian.

## 10.2 MEAN-FIELD THEORETIC ANALYSIS

In this section, we approximate the lumped process defined in Section 10.1.3, when  $M$  is large. Mean-field theory is extensively used for this purpose (Durrett 2010b; Lelarge and Bolot 2008; Pastor-Satorras and Vespignani 2002). As a first step in this direction, peers are assumed to be independently interacting with a mean environment. This allows us to treat each neighbour of a degree- $k$  peer as an independent sample from a

mean environment. We also impose that peers having the same degree play the same chunk selection strategy and thus, behave indistinguishably in a large random graph, suggesting that such a mean-field behaviour can very well be described by population counts. We, therefore, define a mean-field population model that lumps the original process according to the equivalence relation  $\overset{A}{\sim}$ . We shall index all the relevant quantities by degree  $k$  in the following, instead of indexing by peers.

### 10.2.1 Mean-field master equations

Consider the process  $\{Z_t\}_{t \geq 0}$  defined as  $Z_t := (z_x^k(t) : x \in \mathcal{X}, k \in \mathbb{N})$  where  $z_x^k(t)$  is the number of degree- $k$  peers at buffer configuration  $x \in \mathcal{X}$  at time  $t$ . We get our mean-field transition rates for a degree- $k$  peer as follows, for each  $k \in \mathbb{N}, u \in \mathcal{X}$  and  $i \in [n] \setminus \{1\}$  such that  $u_i = 0$ ,

$$\beta^k(u, u + e_i) = \sum_{l=1}^k \zeta \mathbb{E}[\mathbb{1}(Y_l(i) = 1) \alpha^k(i, u, Y_l)] = k \zeta \mathbb{E}[\mathbb{1}(Y_1(i) = 1) \alpha^k(i, u, Y_1)],$$

where  $\{(Y_l, d_l) \mid Y_l = (Y_l(1), Y_l(2), \dots, Y_l(n)) \in \mathcal{X}, d_l \in \mathbb{N}\}_{l=1}^k$  is a set of  $k$  iid samples from the mean environment of a degree- $k$  peer. The first component of each neighbour is the buffer state and the second component, its degree. Note that  $d_l$ 's are distributed according to  $q$  of (10.1.1). Then,

$$\begin{aligned} \mathbb{E}[\mathbb{1}(Y_1(i) = 1) \alpha^k(i, u, Y_1)] &= \sum_{v \in \mathcal{X}: v_i=1} \sum_{m \in \mathbb{N}} \alpha^k(i, u, v) \mathbb{P}(Y_1 = v \mid d_1 = m) \mathbb{P}(d_1 = m) \\ &= \sum_{v \in \mathcal{X}: v_i=1} \sum_{m \in \mathbb{N}} q(m) \frac{\mathbb{E}[z_v^m]}{n_m} \alpha^k(i, u, v). \end{aligned}$$

where  $n_m$  is the number of peers of degree  $m$ . Thus, we get,

$$\beta^k(u, u + e_i) = k \zeta \sum_{v \in \mathcal{X}: v_i=1} \sum_{m \in \mathbb{N}} q(m) \frac{\mathbb{E}[z_v^m]}{n_m} \alpha^k(i, u, v), \quad (10.2.1)$$

for each  $k \in \mathbb{N}$ ,  $u \in \mathcal{X}$  and  $i \in [n] \setminus \{1\}$  such that  $u_i = 0$ . For  $i = 1$ , we set  $\beta$  such that  $\sum_{u \in \mathcal{X}: u_1=1} \frac{z_u^k - e_1}{n_k} \beta^k(u - e_1, u) = 1/M$ , the total input to the system by the server. Now, to write down the master equation, we need to define the change vector  $\varrho : \mathbb{N} \times \mathcal{X} \times [n] \rightarrow \{-1, 0, 1\}^{|\mathcal{X}| \times \mathbb{N}}$  such that  $Y = Z - \varrho(k, u, i) \implies y_u^k = z_u^k + 1, y_{u+e_i}^k = z_{u+e_i}^k - 1, y_x^l = z_x^l \forall l \in \mathbb{N} \setminus \{k\}, x \in \mathcal{X} \setminus \{u\}$  (note that the count vector sums up to the number of peers in the system at all times). Broadening the scope of definition of  $\beta$  by setting it to 0 for all  $u, u + e_i$  not covered in (10.2.1), for large  $M$ , we have the following mean-field CME,

$$\begin{aligned} \frac{d}{dt} \mathbb{P}(Z) &= -\mathbb{P}(Z) + \sum_{\substack{Y: \sum_{v=u} y_v^l = z_u^l \\ \forall u, v \in \mathcal{X}, l \in \mathbb{N}}} \left[ \mathbb{P}(Y) + \sum_{l \in \mathbb{N}, u \in \mathcal{X}, i \in [n]} (y_u^l + 1) \beta^l(u, u + e_i) \times \mathbb{P}(Y - \varrho(l, u, i)) \right. \\ &\quad \left. - \sum_{l \in \mathbb{N}, u \in \mathcal{X}, i \in [n]} y_u^l \beta^l(u, u + e_i) \mathbb{P}(Y) \right]. \end{aligned} \quad (10.2.2)$$

The terms on right hand side correspond to the influx and outflux of probabilities of observing a particular counts' vector.

In order to study the mean dynamics of the count vector  $Z$ , we begin by first setting  $P(Y) = 0 \forall Y \notin \mathbb{N}_0^{|\mathcal{X}| \times \mathbb{N}}$  where  $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ , and then by defining, for each  $l \in \mathbb{N}, u \in \mathcal{X}, i \in [n]$ , the following quantity  $\gamma_{l,u,i}(Z) := z_u^l \beta^l(u, u + e_i)$ . Next, we note that, in mean field, we can write  $E[\gamma_{l,u,i}(Z)]$  as  $E[z_u^l] \beta^l(u, u + e_i)$ . The following result encapsulates the mean dynamics of the system.

**Result 10.2.1.** *The process  $\{Z_t\}_{t \geq 0}$  admitting the mean -field CME (10.2.2) satisfies*

$$\frac{d}{dt} E[Z] = -E[Z] + E[Y] + \sum_{l \in \mathbb{N}, u \in \mathcal{X}, i \in [n]} \varrho(l, u, i) E[\gamma_{l,u,i}(Y)], \quad (10.2.3)$$

where  $Y \in \mathbb{N}_0^{|\mathcal{X}| \times \mathbb{N}}$ , encapsulating the state transitions due to shifting, is such that  $y_u^l = \sum_{v=u} z_v^l \forall l \in \mathbb{N}, u \in \mathcal{X}$ .

We make use of the following lemma to prove Result 10.2.1. The lemma tells us that some calculations required for the mean dynamics get automatically simplified because of the shifting operator.

**Lemma 10.2.1.** *For  $Y$  as defined in Result 10.2.1, the following identity holds true, for all  $k \in \mathbb{N}$ ,*

$$\sum_{Z \in \mathbb{N}_0^{|\mathcal{X}| \times \mathbb{N}}} z_u^k \sum_{Y: \sum_{v=u} y_v^l = z_u^l \forall u, v \in \mathcal{X}, l \in \mathbb{N}} P(Y) = \sum_{v \in \mathcal{X}: \sum_{v=u} z_v^k} E[z_v^k].$$

The proofs of Lemma 10.2.1 and Result 10.2.1 are provided in Appendix H.2. Looking closely at (10.2.3) and recalling the definition of  $\varrho(l, u, i)$ , we write down explicitly, for each  $u \in \mathcal{X}, k \in \mathbb{N}$

$$\frac{d}{dt} E[z_u^k] = -E[z_u^k] + \sum_{v \in \mathcal{X}: \sum_{v=u} z_v^k} \left[ E[z_v^k] + \sum_{i \in [n]} E[z_{v-e_i}^k] \beta^k(v - e_i, v) - \sum_{i \in [n]} E[z_v^k] \beta^k(v, v + e_i) \right], \quad (10.2.4)$$

a self-consistent (autonomous) set of ODEs for the mean population counts.

It is convenient to work with proportions to study the mean dynamics. Therefore, define  $W_t := (w_x^k(t) : x \in \mathcal{X}, k \in \mathbb{N})$  where  $w_x^k(t) := z_x^k / n_k$ . We argue that, when the number of peers is large, it suffices to study the mean dynamics of the proportions, for the fluctuation around mean is expected to be negligible for large systems (Kurtz 1981). Therefore, denoting  $E[w_x^k]$ , with abuse of notation, by  $w_x^k$  itself, we write down the following *rate equations*,

$$\frac{d}{dt} w_u^k = -w_u^k + \sum_{v \in \mathcal{X}: \sum_{v=u} z_v^k} \left[ w_v^k + \sum_{i \in [n]} \left( w_{v-e_i}^k \beta^k(v - e_i, v) - w_v^k \beta^k(v, v + e_i) \right) \right], \quad (10.2.5)$$

for each  $u \in \mathcal{X}, k \in \mathbb{N}$ . We find steady-state proportions by setting  $\frac{d}{dt} w_u^{(k)} = 0$ , giving rise to following fixed point equations at steady state,

$$w_u^k = \sum_{v \in \mathcal{X}: \sum_{v=u} z_v^k} \left[ w_v^k + \sum_{i \in [n]} \left( w_{v-e_i}^k \beta^k(v - e_i, v) - w_v^k \beta^k(v, v + e_i) \right) \right]. \quad (10.2.6)$$



Observe that  $\sum_{u \in \mathcal{X}} \frac{d}{dt} w_u^k = 0$  for all  $k \in \mathbb{N}$ . This is because the proportions sum up to 1, i.e.,  $\sum_{u \in \mathcal{X}} w_u^k = 1 \forall k \in \mathbb{N}$ . We use the above fixed point equations to study the buffer probabilities, which are our performance metrics. However, before doing that, we offer the following remark regarding the interesting connection between our peer-to-peer live streaming model and an infection model from stochastic epidemiology literature.

**Remark 10.2.1** (Connection to infection models). It merits attention that the population model presented here can be thought of as an infection model with  $2^n$  distinct levels of a disease, each level being represented by a  $u \in \mathcal{X}$  and (gradual) recovery being represented by the shifting of buffer state after playback. This amounts to saying, a peer with all buffer positions filled is infected to the highest extent of a disease and if it does not download any chunk, i.e., if it does not get infected, it will gradually recover to a state of complete susceptibility (no chunk available).

**PERFORMANCE METRICS** One of the key metrics of performance in live streaming context is the buffer probability. The buffer probability of index  $i$  of a degree- $k$  peer is the probability that a degree- $k$  peer has a chunk at buffer index  $i$ . In mean field, this becomes the proportion of degree- $k$  peers that have chunks at buffer index  $i$ . Therefore, we define  $p_k : [n] \rightarrow [0, 1]$ , the buffer probability of a peer of degree  $k \in \mathbb{N}$  as

$$p_k(i) = \sum_{u \in \mathcal{X}: u_i=1} w_u^k. \quad (10.2.7)$$

The corresponding global performance of the network is linked to these degree-specific buffer probabilities through the associated degree distribution of  $G$  as follows

$$p(i) = \sum_{k \in \mathbb{N}} \pi(k) p_k(i). \quad (10.2.8)$$

Our goal is to devise a chunk scheduling strategy that optimises these two performance metrics. In order to do so, we derive a recurrence relation among  $p_k$ 's by means of (10.2.6) to understand their behaviour. We have the following result in that direction.

**Result 10.2.2.** *The process  $\{W_t\}_{t \geq 0}$  of proportions obeying rate equation (10.2.5), admits the following recursion relation among the buffer probabilities at steady state*

$$\begin{aligned} p_k(i+1) &= p_k(i) + \sum_{u \in \mathcal{X}: u_i=1} w_{u-e_i}^k \beta^k(u - e_i, u), \\ p(i+1) &= p(i) + \sum_{k \in \mathbb{N}} \pi(k) \sum_{u \in \mathcal{X}: u_i=1} w_{u-e_i}^k \beta^k(u - e_i, u), \end{aligned}$$

for all  $i, k \in \mathbb{N}$ . Moreover, buffer probabilities are non-decreasing functions of their arguments, i.e., buffer indices.

The proof of Result 10.2.2 becomes easier in light of the following lemma that establishes two identities.

**Lemma 10.2.2.** *For the process  $\{W_t\}_{t \geq 0}$  obeying rate equation (10.2.5), for each  $i, k \in \mathbb{N}$ , we have the following two identities*

$$\sum_{u \in \mathcal{X}: u_{i+1}=1} \sum_{v \in \mathcal{X}: v=u} w_v^k = p_k(i),$$

$$\sum_{u \in \mathcal{X}: u_i=1} \sum_{j \in [n]} \left[ \lambda^k(u - e_j, u) - \lambda^k(u, u + e_j) \right] = \sum_{u \in \mathcal{X}: u_i=1} \lambda^k(u - e_i, u).$$

The proofs of Result 10.2.2 and Lemma 10.2.2 are in Appendix H.2.

**Remark 10.2.2** (Interpretation of Result 10.2.2). The left hand side of the recurrence relation gives the probability that the chunk required to fill the buffer location  $i + 1$  is present. The right hand side tells us that there are two possible ways to have the chunk at buffer index  $i + 1$  present. First, it could already be there at buffer index  $i$ , with probability of buffer index  $i$ , and was made available at index  $i + 1$  due to shifting. Second, the chunk was not there, but the peer could download it in the mean time. Roughly speaking, this occurs with probability  $\sum_{u \in \mathcal{X}: u_i=1} w_{u-e_i}^k \beta^k(u - e_i, u)$  for a degree- $k$  peer. This forms the basis of our further analysis of buffer probabilities.

Now we make use of a largely adopted assumption about the chunk selection function. We assume that the chunk selection function of a degree- $k$  peer,  $\alpha^k(i, u, v)$  does not depend on any particular value of  $u$  and  $v$ , but rather assigns probability to buffer indices according to their relative importance as pronounced by EDF and LDF. Call this simplified policy  $s_k$ , instead of  $\alpha^k$ . This implies,

$$\beta^k(u, u + e_i) = k\zeta \sum_{v \in \mathcal{X}: v_i=1} \sum_{l \in \mathbb{N}} q(l) w_v^l \alpha^k(i, u, v) = k\zeta s_k(i) \sum_{l \in \mathbb{N}} q(l) p_l(i) = k\zeta s_k(i) \theta_i,$$

where  $i \in [n]$  and  $\theta_i := \sum_{l \in \mathbb{N}} q(l) p_l(i)$  encapsulates the probability that an arbitrarily given edge points to a node where chunk  $i$  is available.

Let us now revisit the recurrence relation in Result 10.2.2 and plug in the above simplified quantities. In order to do so, note that, for all  $i \in [n]$ ,

$$\begin{aligned} \sum_{u \in \mathcal{X}: u_i=1} w_{u-e_i}^k \beta^k(u - e_i, u) &= \sum_{v \in \mathcal{X}: v_i=0} w_v^k \beta^k(v, v + e_i) \\ &= k\zeta \theta_i s_k(i) \sum_{v \in \mathcal{X}: v_i=0} w_v^k = k\zeta \theta_i (1 - p_k(i)) s_k(i). \end{aligned}$$

The recursion relation in Result 10.2.2 then reads

$$p_k(i + 1) = p_k(i) + k\zeta \theta_i (1 - p_k(i)) s_k(i), \quad (10.2.9)$$

where  $k \in \mathbb{N}$ ,  $i = 1, 2, \dots, n - 1$ , and  $\varphi := p_k(1) = 1/M$ . Such a recurrence relation in the special case of a homogeneous system has served as a starting point for the study of buffer probabilities in a number of articles in the literature, e.g., Ying, Srikant, and Shakkottai (2010), Zhou, D. M. Chiu, et al. (2007), and Zhou, D.-M. Chiu, and Lui (2011). In fact, by choosing  $\pi(k) = \mathbb{1}(k = k^*)$ ,  $\zeta = \frac{1}{k^*}$  for some  $k^* \in \mathbb{N}$ , we retrieve from (10.2.9) the corresponding recurrence relation in the homogeneous set-up, as found in

Ying, Srikant, and Shakkottai (2010), Zhou, D. M. Chiu, et al. (2007), and Zhou, D.-M. Chiu, and Lui (2011). Our endeavour was to provide a principled approach to derive such a recurrence relation in a more general heterogeneous set-up exhibiting degree dependence of peers.

**Remark 10.2.3.** The equations (10.2.9), and (10.2.8) are two key instruments in our analysis of buffer probabilities. While (10.2.9) describes the playback experience of a degree- $k$  peer, a local aspect, (10.2.8) allows us to combine these local information through degree distributions of arbitrary networks to give us a global view. This is notable because even this simple, approximate model allows us to capture the dependence of performance on network structure by plugging in its degree distribution.

We now focus on the two popular chunk selection strategies, namely, LDF and EDF. We follow the same interpretations of EDF and LDF as laid down in Zhou, D.-M. Chiu, and Lui (2011). Also see Figure 10.1.

### 10.2.2 Chunk selection function

#### 10.2.2.1 Latest deadline first strategy

This strategy aims to download the rarest piece first. The priority is thus on the initial buffer indices. Therefore,  $s_k(i)$  can be written as

$$s_k(i) = [1 - \varphi] \prod_{j=1}^{i-1} [p_k(j) + (1 - p_k(j))(1 - k\zeta\theta_j)].$$

The explanation is simple and is provided in KhudaBukhsh, Rückert, et al. (2015). This gives us the following result.

**Result 10.2.3.** 1. The chunk selection function for the LDF strategy can be expressed as

$$s_k(i) = 1 - p_k(i). \quad (10.2.10)$$

2. The recursion relation for buffer probabilities for the LDF strategy has the following form, for  $i = 1, 2, \dots, n-1$  and  $k \in \mathbb{N}$

$$p_k(i+1) = p_k(i) + k\zeta\theta_i(1 - p_k(i))^2. \quad (10.2.11)$$

The proof is similar to Zhou, D.-M. Chiu, and Lui (2011), however, for the sake of completeness, it is provided in KhudaBukhsh, Rückert, et al. (2015).

#### 10.2.2.2 Greedy strategy

The greedy strategy or the EDF strategy seeks to download pieces that are close to playback. The priority is thus on playback urgency and hence on the final buffer indices. Therefore, the chunk selection function can be expressed as

$$s_k(i) = [1 - \varphi] \prod_{j=i+1}^{n-1} [p_k(j) + (1 - p_k(j))(1 - k\zeta\theta_j)].$$

The explanation is similar to the case of the LDF strategy, with the notable exception that now we require to search buffer index  $n$  first, then  $n - 1$  and so on.

**Result 10.2.4.** 1. The chunk selection function for the greedy strategy can be expressed as

$$s_k(i) = 1 - \varphi - p_k(n) + p_k(i + 1). \quad (10.2.12)$$

2. The recursion relation for buffer probabilities for the greedy strategy has the following form, for  $i = 1, 2, \dots, n - 1$  and  $k \in \mathbb{N}$

$$p_k(i + 1) = p_k(i) + k\zeta\theta_i(1 - p_k(i)) [1 - \varphi - p_k(n) + p_k(i + 1)]. \quad (10.2.13)$$

The proof is provided in KhudaBukhsh, Rückert, et al. (2015).

**Remark 10.2.4.** A typical EDF buffer probability curve exhibits a late, sharp increase, contrary to an LDF curve (see Zhou, D. M. Chiu, et al. (2007) and Zhou, D.-M. Chiu, and Lui (2011)). However, when  $M$  is large, EDF hinders propagation of new chunks. While LDF is known to possess good scalability, EDF outperforms LDF when  $M$  is small. We wish to exploit this feature of EDF even when  $M$  is large. In order to do so, we must devise a way to arrest the content bottleneck. We conjecture that this can be done by employing a reasonably small percentage of strong peers (the ones with higher bandwidth, say, but not necessarily connected directly to the server) to play LDF so as to act as *pseudo-servers* in the system. We pursue this idea by studying different strategy profiles in a minimal set-up with only two degrees, where we call the peers of higher degree strong peers and peers of smaller degree, weak peers.

### 10.2.3 A two-degree system

Suppose there are only two degrees  $k_1, k_2 \in \mathbb{N}$  in the system where  $k_1 < k_2$ . For typographical convenience, we shall subscript all the relevant variables with only 1, 2 instead of  $k_1, k_2$  respectively, whenever the degree of a vertex appears as a subscript or as an argument to a function, e.g.,  $\pi_1, \pi_2$  in place of  $\pi(k_1), \pi(k_2)$  respectively and  $p_1(i), p_2(i)$  in place of  $p_{k_1}(i), p_{k_2}(i)$  respectively.

#### Pure LDF strategy

As seen in Section 10.2.2.1, buffer probabilities for the two degrees  $k_1, k_2$  when everybody plays LDF, are given by the following recursion relations

$$p_1(i + 1) = p_1(i) + k_1\zeta\theta_i(1 - p_1(i))^2, \quad (10.2.14)$$

$$p_2(i + 1) = p_2(i) + k_2\zeta\theta_i(1 - p_2(i))^2, \quad (10.2.15)$$

for  $i = 1, 2, \dots, n - 1$ . We adopt a continuous approximation of the above two difference equations (as done in Ying, Srikant, and Shakkottai (2010) and Zhou, D.-M. Chiu, and Lui (2011), for instance). Treating the buffer index  $i$  as a continuous variable  $x$  and

writing  $y_1, y_2, \theta$  for  $p_1(i), p_2(i)$  and  $\theta_i$  respectively, we have the following differential equations

$$\frac{d}{dx}y_1 = k_1\zeta\theta(1-y_1)^2, \quad \frac{d}{dx}y_2 = k_2\zeta\theta(1-y_2)^2. \quad (10.2.16)$$

$$(10.2.17)$$

The above allows an exact solution which we present in the next result.

**Result 10.2.5.** *For the pure LDF strategy and large systems, i.e., when  $M \rightarrow \infty$ , the two buffer probabilities are related according to the following equation*

$$y_2 = \frac{y_1}{r + (1-r)y_1}, \quad (10.2.18)$$

where  $r = \frac{k_1}{k_2}$  is the relative strength of the weak peers compared to the strong ones.

The proof is given in KhudaBukhsh, Rückert, et al. (2015). We immediately see that  $y_2 > y_1$ , i.e., the stronger peers have better performance owing to their greater rate of interaction. However, this difference in performance for the weak peers due to degree disparity can be made arbitrarily small if a sufficiently large buffer is made available. Another interesting consequence is that the above can now be used to derive an expression for buffer-size requirements and facilitate sensitivity analysis therefrom. That is, given  $\epsilon_1 = 1 - p_1(n)$ , the playback discontinuity of the weak peers, we can find the required buffer length of the weak peers  $n_1 = f(\pi, r, \epsilon_1)$  that ensures performance at level  $\epsilon_1$  for some  $f^3$ . Notice that the global performance is related to  $\epsilon_1$  by

$$1 - \epsilon = \pi_1(1 - \epsilon_1) + \pi_2 \frac{1 - \epsilon_1}{1 - (1-r)\epsilon_1},$$

where  $1 - \epsilon = p(n)$ . This can be used when we intend to achieve a pre-specified level of global performance.

*Mixed strategy: SCHEDMIX*

Now we turn to the mixed strategy referred to as SCHEDMIX. Suppose the weaker peers of degree  $k_1$  adopt EDF and the stronger peers of degree  $k_2$ , LDF. Following Sections 10.2.2.1 and 10.2.2.2, we have the following recursion relations

$$p_1(i+1) = p_1(i) + k_1\zeta\theta_i(1-p_1(i)) [1 - \varphi - p_1(n) + p_1(i+1)], \quad (10.2.19)$$

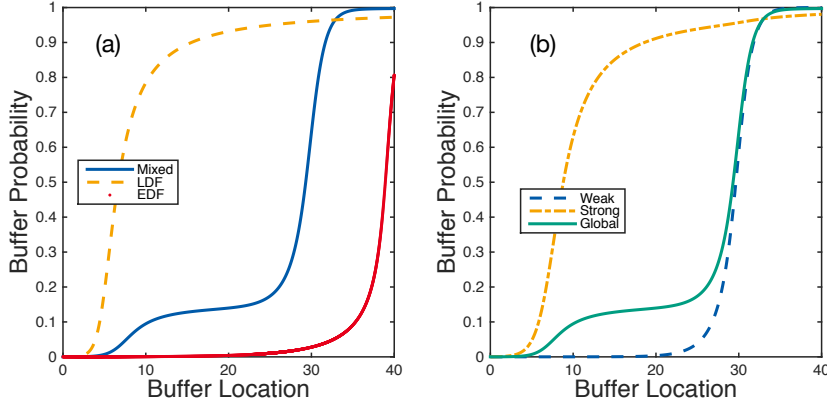
$$p_2(i+1) = p_2(i) + k_2\zeta\theta_i(1-p_2(i))^2, \quad (10.2.20)$$

for  $i = 1, 2, \dots, n-1$ . As before, we shall use a continuous approximation to study their behaviour. Writing  $\epsilon_1 = 1 - p_1(n)$ , we get the following differential equations:

$$\frac{d}{dx}y_1 = \frac{k_1\zeta\theta(1-y_1)(y_1 - \varphi + \epsilon_1)}{1 - k_1\zeta\theta(1-y_1)}, \quad \frac{d}{dx}y_2 = k_2\zeta\theta(1-y_2)^2. \quad (10.2.21)$$

$$(10.2.22)$$

<sup>3</sup> The exact expression is provided in KhudaBukhsh, Rückert, et al. (2015).



**Figure 10.2:** Performance comparison based on mean-field analysis of buffer probabilities. (a) Global buffer probabilities for the three strategy profiles. SCHEDMIX gives higher playback continuity than both EDF and LDF for the given buffer length. (b) Comparison of weak versus strong under SCHEDMIX. Weak peers indeed eventually outperform the strong peers under SCHEDMIX. Parameter values:  $M = 10000, k_1 = 5, k_2 = 15, \pi_1 = 0.85 = 1 - \pi_2, \zeta = 0.20$ .

The above equations, unfortunately, do not yield an analytic solution. Therefore, we resort to numerical solution to compare global performance of the system under different strategy profiles. It turns out that performance under SCHEDMIX is indeed better than that under the pure LDF strategy (see Figure 10.2), substantiating our claim. In Appendix H.3, we also provide a game theoretic justification in favour of SCHEDMIX.

When we compared performance of weak peers versus strong ones, an interesting phenomenon was observed. The weak peers could eventually manage to outperform the strong ones, caused by a sharp increase in buffer probabilities that a typical “EDF curve” enjoys and what we call the boon of heterogeneity (see Figure 10.2). This phenomenon is in agreement with our supposition and can be explained intuitively. Both strong and weak peers benefit from being exposed to a heterogeneous environment. In a homogeneous set-up, one would expect somewhat similar availability of chunks among all its neighbours. On the contrary, a heterogeneous environment makes available a diverse collection of chunks. This prepones the steep rise that a typical “EDF curve” enjoys. Since an EDF curve has a greater growth-rate in the neighbourhood of 1 (see Zhou, D. M. Chiu, et al. (2007) and Zhou, D.-M. Chiu, and Lui (2011)), weak peers can eventually outperform LDF-playing strong peers even for moderate buffer-lengths.

**Remark 10.2.5.** We do not consider the pure EDF strategy separately here as it can be studied in a similar fashion. In KhudaBukhsh, Rückert, et al. (2015), we also provide a short stability analysis that gives an additional justification of why the weak peers outperform the strong ones.

### 10.3 SIMULATION RESULTS

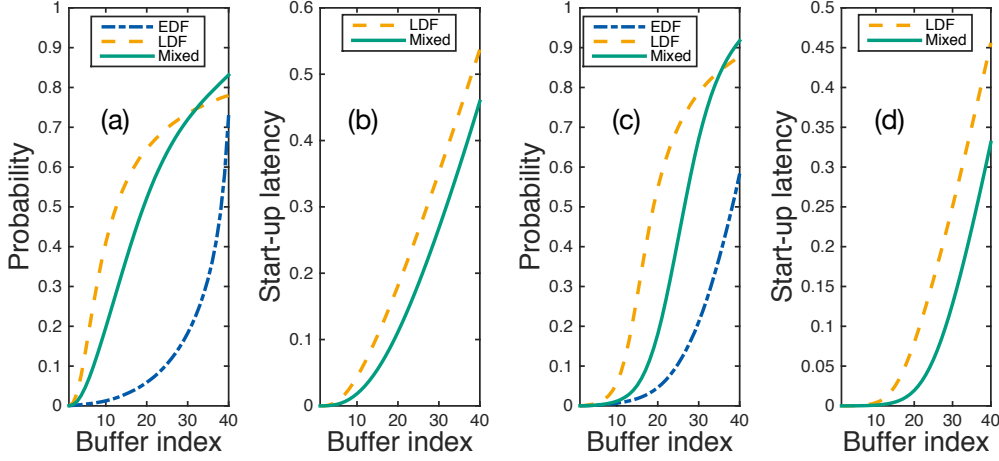
In this section, we document our findings from the simulation of the stochastic model. This is carried out in two steps: first, generation of a random graph and second, simulation of the content delivery process in accordance with Section 10.1.

**START-UP LATENCY** The second metric that we look at is the start-up latency. It is the time a peer should wait before starting playback. While there is no unanimity as to how one should define this quantity, it is reasonable to wait until a newly arrived peer's buffer attains a steady state. If it starts playback before that, it is likely to experience below steady state playback quality initially. On the other hand, waiting longer will not improve long-term playback experience. In a homogeneous set-up where everybody plays the same policy and has the same buffer probabilities, as argued in Zhou, D. M. Chiu, et al. (2007), this is well represented by  $\sum_i p(i)$ , the average number of available chunks at each peer. In our heterogeneous model, a higher degree peer interacts more often than a lower degree peer. Therefore, a newly arrived degree- $k$  peer should have start-up latency of  $k\zeta \sum_i p(i)$  in the mean-field. The corresponding global metric follows as  $E[k]\zeta \sum_i p(i)$ . For aesthetic reasons, we normalise this quantity to  $(0, 1)$ .

**IMPACT OF NETWORK STRUCTURE** In order to see the impact of network structure, we perform simulation of the model on BA preferential attachment (Barabási and Albert 1999) and WS small world (Watts and Strogatz 1998) networks. Simulation results on a BA network with 2000 peers (with 25% of them playing LDF) and that on a WS network with 5000 peers (with 20% of them playing LDF) are depicted in Figure 10.3. In both cases, the mixed strategy SCHEDMix gives a better performance, corroborating our claim. More importantly, it causes a significant reduction in start-up latency.

**Remark 10.3.1.** Although Figure 10.3 affirms that SCHEDMix does outperform the pure LDF and the pure EDF strategies, the crux of employing SCHEDMix remains in letting most peers play greedy. SCHEDMix, thus, allow for smaller start-up latency to ensure good playback performance for everyone (at least as good as pure LDF strategy). This is a significant benefit.

**EXTENSION TO OTHER CENTRALITY MEASURES** The idea behind SCHEDMix is simple: exploit the capabilities of the strong peers to help the weak ones. SCHEDMix achieves this through degree-based assignment of strategies, but the notion goes beyond degrees. The virtues of SCHEDMix can also be achieved by taking into account other important networking factors, such as betweenness centrality, well-connectedness to the server. For example, in case of WS graphs, betweenness centrality better captures the notion of strength than degrees. To demonstrate the idea, we performed a betweenness centrality-based strategy assignment. In this variant of SCHEDMix, nodes having higher betweenness centrality are assigned LDF and all others, EDF. In Figure 10.4 we show that this variant of SCHEDMix also outperforms pure LDF and EDF strategies in a WS graph with 2000 nodes (with 25% playing LDF).



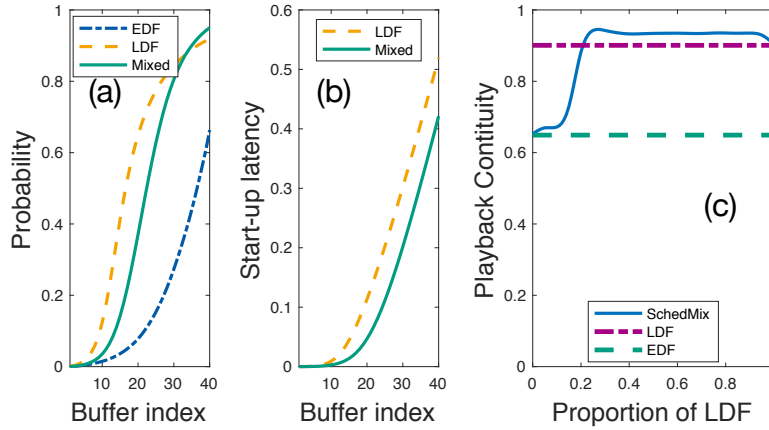
**Figure 10.3:** Impact of network structure and performance evaluation in terms of buffer probabilities and the start-up latency on a BA and a WS graph. Figures (a), (b) show performance on a BA graph with 2000 peers. Figures (c), (d) display performance on a WS graph with 5000 peers. In both cases,  $n = 40, \zeta = 0.25$ . Please note that start-up latency is shown only for strategies ensuring playback continuity of at least 0.75 with buffer size  $n = 40$ .

**OPTIMAL THRESHOLD** One of the main advantages of SCHEDMIX is that it requires only a small percentage of strong peers to play LDF in order to uplift the weak peers and improve overall playback experience. However, the optimal percentage of strong peers required to do so will depend on various factors, and in general, is a non-trivial question. In Figure 10.4 we study how the overall playback experience changes as we change the proportion of nodes playing LDF in SCHEDMIX for WS graphs with 2000 nodes. Strategy assignment is carried out as per betweenness centrality. It is interesting to observe that SCHEDMIX outperforms pure strategies over a broad range of strategy assignment, allowing greater freedom when designing scheduling strategies in practical applications. It also substantiates the boon of heterogeneity phenomenon.

#### 10.4 DISCUSSION

Our mathematical framework can also serve as a foundation in problems other than the one in pursuit, *e.g.*, network security problems such as circulation of updates to anti-virus in the event of cyber attacks or the circulation of virus/malware itself, supply chain problems for products with limited validity, express consignment delivery problems. Its shifting feature makes it particularly interesting as it allows for multiple interpretations, *e.g.*, advertisement of promotional offers with deadlines, gradual recovery or mutation in the context of infection spread. Keeping analytic tractability aside, the prospect of incorporating more sophisticated mechanisms in practical implementation is broad. We expect to see application of SCHEDMIX in combination with more sophisticated mech-





**Figure 10.4:** In Figures (a) and (b), we compare of different strategy profiles under betweenness centrality-based SCHEDMIX. In Figure (c), we study how the overall playback continuity behaves as a function of the proportion of LDF-playing peers. Interestingly, SCHEDMIX outperforms pure strategies over a broad range of strategy assignment. It also substantiates the boon of heterogeneity phenomenon.

anisms. One straightforward but important step is the application of SCHEDMIX in a state-of-the-art hybrid streaming system, where both mesh/pull and multi-tree/push-based mechanisms coexist. In this context it would also be interesting to understand the impact of other mechanisms, such as exchange of buffermaps or a streaming of layered media content. However, as a recent work by Silva, Dias, and Ricardo (2016) shows, avoiding knowledge on chunk availability can be desired to preserve the privacy of users, making the streaming approach without buffermaps assumed in this work of particular interest. The results presented in this work are encouraging in that SCHEDMIX could be used as an alternative to complex scheduling strategies in the growing number of scenarios where peer heterogeneity is inevitably given, *e.g.*, when bandwidth-constrained mobile users meet well-connected and high-capacity home users. Besides, the results could be used in the planning of *transitions* (Frömmgen et al. 2015) between strategies when environmental conditions change.

In this chapter, we made the dependence of performance on degree of the vertices explicit. The idea of a degree-based (strength-based) combination of primitive scheduling strategies led to two interesting revelations, namely, the boon of heterogeneity and the weak peers outperforming the strong ones. Inspired by these observations, we proposed our mixed strategy SCHEDMIX. In the next and final chapter of this dissertation, we shall discuss future research directions. A summary of the author’s contributions will also be provided.



## CONCLUDING REMARKS

---

### 11.1 SUMMARY OF CONTRIBUTIONS

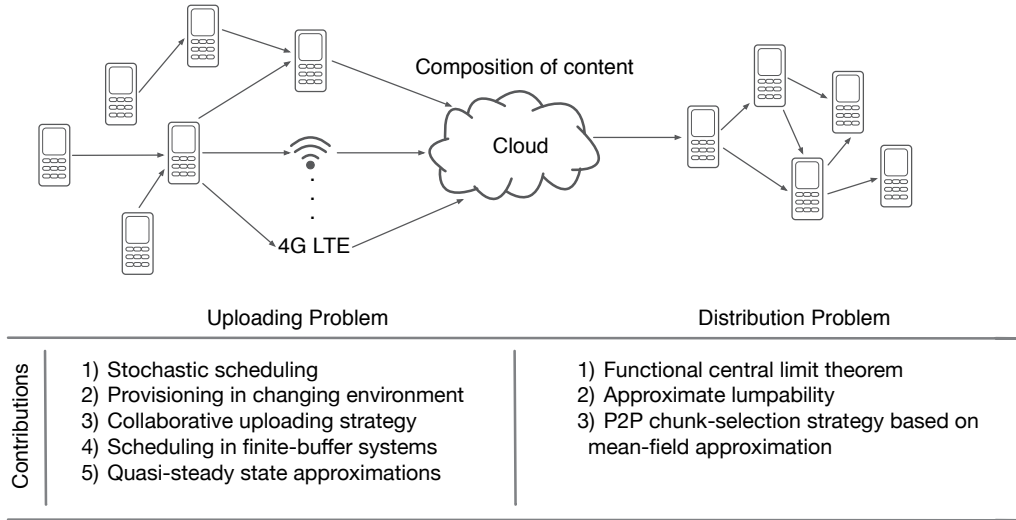
The present dissertation considered two different classes of models from applied probability literature, namely, parallel queueing systems and Markovian Agent-based Models. The research questions arose, to a large extent, from a communication networking scenario presented in Section 1.1 (see also Figures 1.1 and 11.1). On one hand, we have heterogeneous devices that collaboratively upload certain content to the cloud. On the other hand, we have distribution of content from the cloud (after necessary aggregation, composition and personalisation of content) to end-users.

#### 11.1.1 *Uploading problem*

The *uploading problem* is an umbrella term for research problems arising from the uploading leg of the scenario and is studied in detail in Chapters 3 to 7.

For the intermittent uploading case, the ability to calculate the performance metrics in closed-form using the theory of order statistics facilitates designing uploading strategies and their performance evaluation. Based on the closed-form expressions, optimal uploading strategies were devised in Chapter 5. Modelling the stream uploading problem as a scheduling problem in a parallel queueing system, we developed the notions of stochastic scheduling (Chapter 3), and provisioning (Chapter 4) in FJ queueing systems. The objectives of designing optimal stochastic schedules and provisions were achieved by approximating probabilities of rare events with exponential estimates making use of martingale techniques and establishment of a Large Deviations Principle. The resultant theoretical insights were finally used to design optimal collaborative uploading strategies (Chapter 5).

Two special subclasses of queueing systems are considered additionally. The first class constitutes of queueing systems with finite buffers (Chapter 6). We used the random time change representation for Markov processes for this purpose and discussed preliminary ideas on optimal probabilistic scheduling in such systems. We also derived a scaling limit as the number of servers increases to infinity. The second special subclass has a close resemblance to the Michaelis-Menten enzyme kinetics (Chapter 7). We derived various Quasi-Steady State Approximations directly from the stochastic description of the system using multi-scaling techniques from probability literature. In particular, we considered the standard QSSA, the total QSSA, and the reversible QSSA. In the context of the communication system presented in Section 1.1, this approximation is useful in situations such as when the number of packets to be transmitted is too large compared to the number of paths available.



**Figure 11.1:** Summary of contributions of the author in the light of the telecommunication scenario presented in Section 1.1 (see also Figure 1.1).

The *uploading problem* is an umbrella term for research problems arising from the uploading leg of the scenario. Modelling the uploading problem as a scheduling problem in a parallel queueing systems, the author made contributions in developing notions of stochastic scheduling, provisioning in FJ queueing systems. These objectives were achieved by approximating probabilities of rare events with exponential estimates making use of martingale techniques and establishment of LDPs. The resultant theoretical insights were finally used to design optimal collaborative uploading strategies.

The second leg of the scenario, the *distribution problem*, concerns distribution of content from the cloud to end-users. We specifically focus on the large-scale problem when the number of end-users grows arbitrarily large. In order to understand the dynamics of the distribution problem better, we model it as an MABM. Three different approximations are presented in this dissertation. First, an FCLT for key population counts are proved for an information-dissemination process on configuration model random graphs. Second, local symmetry-driven approximate lumpability is studied for a general MABM. Finally, as an application, chunk selection strategies for P2P live streaming systems are thoroughly analysed using mean-field theory and a better mixed strategy, called SCHED-Mix, is proposed.

### 11.1.2 Distribution problem

The second leg of the scenario, the *distribution problem*, concerns distribution of content from the cloud to end-users. We specifically focus on the large-scale problem when the number of end-users grows arbitrarily large. In order to understand the dynamics of the distribution problem better, we model it as an MABM. Three different approximations are presented in this dissertation.

First, an FCLT for key population counts are proved for an ID process on configuration model random graphs in Chapter 8. Special emphasis has been put on Poisson-

type distributions to study the so-called correlation equations approach. Second, local symmetry-driven approximate lumpability is studied for a general MABM. As many large random graphs tend to become asymmetric rendering automorphism-based lumping approach ineffective as a tool of model reduction, we proposed a lumping method based on a notion of local symmetry, which compares only local neighbourhoods of vertices, in Chapter 9. The connections to fibrations of graphs, colour refinements and coverings are also discussed. Finally, as an application, primitive chunk selection strategies for P2P live streaming systems, such as the LDF and the EDF, are thoroughly analysed using mean-field theory and an improved strategy, called SCHEDMIX, is proposed in Chapter 10. SCHEDMIX is shown to outperform the LDF as well as the EDF using a mean-field theoretic analysis of buffer probabilities.

## 11.2 FUTURE DIRECTIONS

### 11.2.1 *Further approximations of queueing systems*

Classical queueing theoretic models have been extended in many directions in the recent times. Many of the nuances of modern technology have already yielded to successful theoretical analysis, but many more of them present challenging research problems that are yet to be solved. Parallelisation is one of the important facets of modern technology and therefore, the class of queueing systems allowing parallelisation was the main focus in this dissertation. In order to be able to devise and employ optimal probabilistic schedules in modern applications, the next natural step would be to extend the results obtained in this dissertation to incorporate other features in the modelling framework.

In the context of FJ queueing systems, accommodating multiple types of customers and devising queue-aware stochastic scheduling algorithms will be very useful from a practical perspective. From the perspective of maintenance of large processing systems (not necessarily of the FJ-type), server repair strategies are very important. It will be interesting to incorporate various repair strategies in the traditional FJ framework and conduct a thorough theoretical performance evaluation.

The analysis of the finite-buffer queueing systems presented in Chapter 6 is far from complete. The probabilistic schedule considered in Chapter 6 is common in queueing systems known as the supermarket model. The preliminary ideas need to be extended to accommodate more sophisticated probabilistic schedules. Some of these extensions are immediately achievable without much theoretical difficulties. In another direction, we can gradually move away from Markovianness towards general processes as much as possible. In particular, computable probability approximations for a  $G/G/N/K$  system allowing a wide range of probabilistic scheduling will be extremely useful.

### 11.2.2 *Queueing theoretic approach to chemical reaction networks*

In Chapter 2, we discussed the connections between queueing theory and chemical reaction networks. In particular, we showed that there is a direct correspondence between some queueing systems, and zero and first order chemical reaction networks. In Chapter 7, we further explored this connection and derived various QSSAs directly from the random time change representations of the species copy numbers for the MM enzyme

kinetic reaction system. The work can be further extended to derive various QSSA for single-stage multiple-server queueing system with multiple classes of customers. Further exploiting the multi-scaling techniques and the random time change representation, we can derive asymptotic limits of various infinite-server queueing systems. In particular, it will be interesting to consider an infinite-server queueing system with multiple classes of customers exhibiting strategic behaviour.

Approximations of the above flavour are an application of tools that have already become standard in the CRN literature to queueing theory. The other direction is also immensely promising. Therefore, a prominent portion of the author's future research effort will be directed towards identifying and consolidating a queueing theoretic approach to CRNs. In particular, the various approximations obtained for non-Markovian queues can be applied to chemical reaction networks.

### 11.2.3 *Further approximations of IPSs*

The MABMs present a number of challenging theoretical problems that will be pursued in the future. The local symmetry-driven approximate lumping discussion presented in Chapter 9 assumes the graph remains unchanged through the course of the dynamics. If we allow the graph to change dynamically, as we may need to in many practical applications, the lumping procedure based on the local symmetries of the static graph will not be effective. Therefore, we need to invent new methods of lumping the states of an MABM when the graph is also allowed to change over time. This approach will be particularly helpful in engineered systems where the graph can be dynamically rewired to our benefit, *e.g.*, to improve performance of the engineered system.

The diffusion approximation proved in Chapter 8 considers the simplest epidemic process, namely the stochastic compartmental SI process, on CM random graphs. This work can be extended in roughly three directions. First, we can extend the results to include other random graph models. Since the CM graphs may not be realistic in many applications, an extension of the results obtained in this dissertation to incorporate other random graph models will be important. The second direction in which the author intends to extend the current work is to incorporate more general processes. In particular, diffusion approximations for other epidemic processes, such as the SIR or the Susceptible-Exposed-Infected-Recovered (SEIR) processes on random graphs with community structures will be important for policy making by public health institutions. The third direction involves extending diffusion approximations to a Large Deviations Principle. This will require significant work because the key quantities of interest, such as the various population counts, the lumped processes etc. are usually not Markovian. Therefore, standard machinery for proving an LDP for Markov processes can not be used directly. However, establishing an LDP for a general IPS is tremendously important in order to accurately estimate probabilities of rare events.

## APPENDICES





## SUPPLEMENTARY MATERIAL TO CHAPTER 3

Before we present our proofs, let us make a remark that will be useful throughout the discourse.

**Remark A.o.1.** If  $\{X_k, \mathcal{F}_k\}$  and  $\{Y_k, \mathcal{F}_k\}$  are submartingales, then  $\{\max(X_k, Y_k), \mathcal{F}_k\}$  are also submartingales (see 7.3.2.(e) *Comments* of Ash (1972) for a proof). Treating martingales as submartingales and extending the above mentioned result to accomodate maximum over a finite collection of submartingales, we can establish that  $\{X(k), \mathcal{F}_k\}_{k \in \mathbb{N}_0}$  is a submartingale whenever  $\{X_n(k), \mathcal{F}(k)\}_{k \in \mathbb{N}_0}$  is a submartingale (or a martingale) for each  $n \in [N]$  and  $X(k) := \max_{n \in [N]} X_n(k), \forall k \in \mathbb{N}_0$ .

## A.1 MAIN PROOFS

*Proof of Theorem 3.1.1.* Notice that the stability condition given in  $\max_{n \in [N]} \mathbb{E}[S_{n,1}] < \mathbb{E}[A_1]$  guarantees the existence of  $\theta_n > 0$  such that  $\alpha_n(\theta_n)\beta(\theta_n) = 1$  for all  $n \in [N]$  (see Boxma, Koole, and Z. Liu (1994) and Poloczek and Ciucu (2014)). Hence,  $\tilde{\theta} > 0$  is well defined. Consider the filtration

$$\mathcal{F}_k := \sigma(\{S_{n,i}\}_{n \in [N], i \leq k}, \{A_i\}_{i \leq k}), \quad (1.1.1)$$

for all  $k \in \mathbb{N}_0$ , where  $\sigma(\{S_{n,i}\}_{n \in [N], i \leq k}, \{A_i\}_{i \leq k})$  denotes the smallest  $\sigma$ -field generated by  $\{S_{n,i}\}_{n \in [N], i \leq k}, \{A_i\}_{i \leq k}$ .

**BOUNDING THE WAITING TIME** For each  $n \in [N]$ , define the stochastic process

$$Z_n(k) := \exp\left(\theta_n \sum_{i=1}^k (S_{n,i} - A_i)\right), \forall k \in \mathbb{N}_0.$$

It can be seen that  $\{Z_n(k), \mathcal{F}_k\}_{k \in \mathbb{N}_0}$  is a martingale. By virtue of the sub- and super-martingale inequalities due to Doob (see Ash (1972, Chapter 7, Problem 3(c))), we have

$$\mathbb{P}\left(\max_{k \in \mathbb{N}_0} Z_n(k) \geq \sigma\right) \leq \frac{\sup_{k \in \mathbb{N}_0} \mathbb{E}[Z_n^+(k)]}{\sigma}, \quad (1.1.2)$$

for  $\sigma \geq 0$  and for each  $n \in [N]$ , where  $Z_n^+(k) := Z_n(k) \wedge 0$ . Now, our martingales are so constructed that  $\sup_{k \in \mathbb{N}_0} \mathbb{E}[Z_n^+(k)] = 1$ . Therefore, we have

$$\mathbb{P}\left(\max_{k \in \mathbb{N}_0} Z_n(k) \geq \sigma\right) \leq \frac{1}{\sigma}. \quad (1.1.3)$$

Now define  $\tilde{\theta} := \min_{n \in [N]} \theta_n$ . Finally, by virtue of *Boole's inequality* and (1.1.3), we bound the tail probabilities of the waiting time  $W$  as follows

$$\begin{aligned}
P(W \geq \sigma) &= P(\max_{n \in [N]} \{ \max_{k \in \mathbb{N}_0} \{ \sum_{i=1}^k (S_{n,i} - A_i) \} \} \geq \sigma) \\
&= P(\cup_{n \in [N]} \{ \max_{k \in \mathbb{N}_0} \{ \sum_{i=1}^k (S_{n,i} - A_i) \} \} \geq \sigma) \\
&\leq \sum_{n \in [N]} P(\max_{k \in \mathbb{N}_0} \{ \sum_{i=1}^k (S_{n,i} - A_i) \} \geq \sigma) \\
&= \sum_{n \in [N]} P(\max_{k \in \mathbb{N}_0} Z_n(k) \geq \exp(\theta_n \sigma)) \\
&= \exp(-\tilde{\theta} \sigma) \sum_{n \in [N]} \exp(-(\theta_n - \tilde{\theta}) \sigma).
\end{aligned}$$

**BOUNDING THE RESPONSE TIME** Define the stochastic process, for each  $n \in [N]$ ,

$$Y_n(k) := \exp(\theta_n (\sum_{i=0}^k S_{n,i} - \sum_{i=1}^k A_i)), \forall k \in \mathbb{N}_0.$$

See that  $\{Y_n(k), \mathcal{F}_k\}_{k \in \mathbb{N}_0}$  is a martingale. Then, by virtue of the sub- and supermartingale inequalities due to Doob (see Ash (1972, Chapter 7, Problem 3(c))), we have

$$P(\max_{k \in \mathbb{N}_0} Y_n(k) \geq \sigma) \leq \frac{\sup_{k \in \mathbb{N}_0} E[Y_n^+(k)]}{\sigma} = \frac{\alpha_n(\theta_n)}{\sigma}, \quad (1.1.4)$$

for  $\sigma \geq 0$  and for each  $n \in [N]$ , because our martingales are so constructed that

$$\begin{aligned}
Y_n^+(k) &:= Y_n(k) \wedge 0 = Y_n(k) \quad \forall k \in \mathbb{N}_0 \\
\implies E[Y_n^+(k)] &= E[\exp(\theta_n S_{n,0})] \prod_{i=1}^k \alpha_n(\theta_n) \beta(\theta_n) = \alpha_n(\theta_n) \quad \forall k \in \mathbb{N}_0.
\end{aligned}$$

Now define  $\tilde{\theta} := \min_{n \in [N]} \theta_n$ . Finally, by virtue of *Boole's inequality* and (1.1.4), we bound the tail probabilities of the response time  $R$  as follows

$$\begin{aligned}
P(R \geq \sigma) &= P(\max_{n \in [N]} \{ \max_{k \in \mathbb{N}_0} \{ \sum_{i=0}^k S_{n,i} - \sum_{i=1}^k A_i \} \} \geq \sigma) \\
&= P(\cup_{n \in [N]} \{ \max_{k \in \mathbb{N}_0} \{ \sum_{i=0}^k S_{n,i} - \sum_{i=1}^k A_i \} \} \geq \sigma) \\
&\leq \sum_{n \in [N]} P(\max_{k \in \mathbb{N}_0} \{ \sum_{i=0}^k S_{n,i} - \sum_{i=1}^k A_i \} \geq \sigma) \\
&= \sum_{n \in [N]} P(\max_{k \in \mathbb{N}_0} Y_n(k) \geq \exp(\theta_n \sigma)) \\
&= \exp(-\tilde{\theta} \sigma) \sum_{n \in [N]} \alpha_n(\theta_n) \exp(-(\theta_n - \tilde{\theta}) \sigma).
\end{aligned}$$

□

*Proof of Theorem 3.2.1.* Consider the filtration

$$\mathcal{F}_k := \sigma(\{S_{n,0}, \tilde{S}_{n,i}\}_{n \in [N], i \leq k}, \{A_i\}_{i \leq k}),$$

for all  $k \in \mathbb{N}_0$ , where  $\sigma(\{S_{n,0}, \tilde{S}_{n,i}\}_{n \in [N], i \leq k}, \{A_i\}_{i \leq k})$  denotes the smallest  $\sigma$ -field generated by  $\{S_{n,0}, \tilde{S}_{n,i}\}_{n \in [N], i \leq k}, \{A_i\}_{i \leq k}$ .

**BOUNDING THE WAITING TIME** For each  $n \in [N]$ , define the stochastic process

$$Z_n(k) := \exp\left(\theta_n \sum_{i=1}^k (\tilde{S}_{n,i} - A_i)\right), \forall k \in \mathbb{N}_0.$$

Analogous to the proof of Theorem 3.1.1, it follows that  $\{Z_n(k), \mathcal{F}_k\}_{k \in \mathbb{N}_0}$  is a martingale. Similarly, it follows

$$\mathbb{P}(\max_{k \in \mathbb{N}_0} Z_n(k) \geq \sigma) \leq \frac{1}{\sigma}. \quad (1.1.5)$$

Now define  $\tilde{\theta} := \min_{n \in [N]} \theta_n$ . Therefore, by *Boole's inequality* and (1.1.5), we bound the tail probabilities of the waiting time  $W$  as follows

$$\begin{aligned} \mathbb{P}(W \geq \sigma) &= \mathbb{P}(\max_{n \in [N]} \{\max_{k \in \mathbb{N}_0} \{\sum_{i=1}^k (\tilde{S}_{n,i} - A_i)\}\} \geq \sigma) \\ &\leq \exp(-\tilde{\theta}\sigma) \sum_{n \in [N]} \exp(-(\theta_n - \tilde{\theta})\sigma). \end{aligned}$$

**BOUNDING THE RESPONSE TIME** Define the stochastic process, for each  $n \in [N]$ ,

$$Y_n(k) := \exp\left(\theta_n(S_{n,0} + \sum_{i=1}^k \tilde{S}_{n,i} - \sum_{i=1}^k A_i)\right), \forall k \in \mathbb{N}_0.$$

Following the proof of Theorem 3.1.1, we show that  $\{Y_n(k), \mathcal{F}_k\}_{k \in \mathbb{N}_0}$  is a martingale, and in particular,

$$\mathbb{P}(\max_{k \in \mathbb{N}_0} Y_n(k) \geq \sigma) \leq \frac{\alpha_n(\theta_n)}{\sigma}. \quad (1.1.6)$$

Now define  $\tilde{\theta} := \min_{n \in [N]} \theta_n$ . Therefore, we bound the tail probabilities of the response time  $R$  as follows using the *Boole's inequality* and (1.1.6),

$$\begin{aligned} \mathbb{P}(R \geq \sigma) &= \mathbb{P}(\max_{n \in [N]} \{\max_{k \in \mathbb{N}_0} \{S_{n,0} + \sum_{i=1}^k \tilde{S}_{n,i} - \sum_{i=1}^k A_i\}\} \geq \sigma) \\ &\leq \sum_{n \in [N]} \mathbb{P}(\max_{k \in \mathbb{N}_0} \{S_{n,0} + \sum_{i=1}^k \tilde{S}_{n,i} - \sum_{i=1}^k A_i\} \geq \sigma) \\ &= \exp(-\tilde{\theta}\sigma) \sum_{n \in [N]} \alpha_n(\theta_n) \exp(-(\theta_n - \tilde{\theta})\sigma). \end{aligned}$$

□

## A.2 STATISTICAL RESULTS

**Lemma A.2.1.** 1. Suppose  $X \sim \text{Binomial}(N, p)$ . Then, the following holds, for  $a > 0$ ,

$$\begin{aligned} \mathbb{E}[Xe^{-aX}] &= \frac{Npe^{-a}}{1-q^N}(pe^{-a}+q)^{N-1} \\ \mathbb{E}[X^2e^{-aX}] &= \frac{Npe^{-a}}{1-q^N}(Npe^{-a}+q)(pe^{-a}+q)^{N-2}. \end{aligned}$$

2. If  $X$  is distributed uniformly over  $[N]$ , then, for  $a > 0$ , the following holds

$$\begin{aligned} \mathbb{E}[Xe^{-aX}] &= \frac{e^{-a}}{N(1-e^{-a})} \left[ \frac{1-Ne^{-(N+1)a}}{(1-e^{-a})} - (N+1)e^{-aN} \right] \\ \mathbb{E}[X^2e^{-aX}] &= \frac{e^{-2a}}{N(1-e^{-a})} \left[ 2 \frac{(1-e^{-(N+1)a})}{(1-e^{-a})^2} \right. \\ &\quad \left. - \frac{2(N+1)e^{-Na} - (1-Ne^{-(N+1)a})}{(1-e^{-a})} - (N+1)(Ne^{-(N-1)a} + e^{-aN}) \right]. \end{aligned}$$

*Proof of Lemma A.2.1.* 1. First note that

$$\mathbb{E}[Xe^{-aX}] = \frac{Npe^{-a}}{1-q^N} \sum_{l \in [N]} \binom{N-1}{l-1} (pe^{-a})^{l-1} q^{(N-1)-(l-1)} = \frac{Npe^{-a}}{1-q^N} (pe^{-a}+q)^{N-1}.$$

Now, see that  $\mathbb{E}[X^2e^{-aX}] = \mathbb{E}[X(X-1)e^{-aX} + Xe^{-aX}] = \mathbb{E}[X(X-1)e^{-aX}] + \mathbb{E}[Xe^{-aX}]$ , where

$$\begin{aligned} &\mathbb{E}[X(X-1)e^{-aX}] \\ &= \frac{N(N-1)(pe^{-a})^2}{1-q^N} \sum_{l \in [N] \setminus \{1\}} \binom{N-2}{l-2} (pe^{-a})^{l-2} q^{(N-2)-(l-2)} \\ &= \frac{N(N-1)(pe^{-a})^2}{1-q^N} (pe^{-a}+q)^{N-2}. \end{aligned}$$

Therefore, we get  $\mathbb{E}[X^2e^{-aX}] = \frac{Npe^{-a}}{1-q^N} (Npe^{-a}+q)(pe^{-a}+q)^{N-2}$ .

2. See that  $\mathbb{E}[Xe^{-aX}] = \frac{e^{-a}}{N(1-e^{-a})} \left[ \frac{1-e^{-(N+1)a}}{(1-e^{-a})} - (N+1)e^{-aN} \right]$ , and  $\mathbb{E}[X^2e^{-aX}] = \mathbb{E}[X(X-1)e^{-aX} + Xe^{-aX}] = \mathbb{E}[X(X-1)e^{-aX}] + \mathbb{E}[Xe^{-aX}]$ . Now,

$$\begin{aligned} \mathbb{E}[X(X-1)e^{-aX}] &= \frac{e^{-2a}}{N(1-e^{-a})} \left[ 2 \frac{(1-e^{-(N+1)a})}{(1-e^{-a})^2} - 2 \frac{(N+1)e^{-Na}}{(1-e^{-a})} \right. \\ &\quad \left. - (N+1)Ne^{-(N-1)a} \right]. \end{aligned}$$

Therefore, we get

$$\begin{aligned} \mathbb{E}[X^2e^{-aX}] &= \frac{e^{-2a}}{N(1-e^{-a})} \left[ 2 \frac{(1-e^{-(N+1)a})}{(1-e^{-a})^2} - \frac{2(N+1)e^{-Na} - (1-e^{-(N+1)a})}{(1-e^{-a})} \right. \\ &\quad \left. - (N+1)(Ne^{-(N-1)a} + e^{-aN}) \right]. \end{aligned}$$

□

## A.3 SERVICE TIME SCALING

*Proof of Theorem 3.3.1.* First note that  $\alpha(u) := \mathbb{E}[e^{uS}] = \frac{\mu}{\mu-u}$  and  $\beta(u) := \mathbb{E}[e^{-uA_1}] = \frac{\lambda}{\lambda+u}$ , whence we find  $\theta = \mu - \lambda > 0$  such that  $\alpha(\theta)\beta(\theta) = 1$ . Since  $g_l(u) = \alpha(\frac{u}{l})$ , the solution to  $g_l(u)\beta(u) = 1$  is given by  $\theta_l := l\mu - \lambda > 0$ .

Now consider the scenario conditional on  $\{L = l\}$  for some  $l \in [N]$ . Proceeding in a similar fashion as in the proof of Theorem 3.1.1 and replacing the probabilities and expectations with the corresponding conditional probabilities and expectations respectively, whenever necessary, we get the following bounds on the conditional tail probabilities of the steady state waiting time and the response time as follows

$$\mathbb{P}(W \geq \sigma \mid \{L = l\}) \leq l e^{-\theta_l \sigma}, \quad \mathbb{P}(R \geq \sigma \mid \{L = l\}) \leq l g_l(\theta_l) e^{-\theta_l \sigma}.$$

Inserting the value of  $\theta_l$ ,

$$\mathbb{P}(W \geq \sigma \mid \{L = l\}) \leq l e^{\lambda \sigma} e^{-\mu \sigma l}, \quad \mathbb{P}(R \geq \sigma \mid \{L = l\}) \leq \frac{e^{\lambda \sigma}}{\rho} l^2 e^{-\mu \sigma l}.$$

Now, to get bounds on the unconditional probabilities, we utilise the above two upper bounds and note that

$$\begin{aligned} \mathbb{P}(W \geq \sigma) &= \sum_{l \in [N]} \mathbb{P}(W \geq \sigma \mid \{L = l\}) \mathbb{P}(L = l) \\ &\leq e^{\lambda \sigma} \sum_{l \in [N]} l e^{-\mu \sigma l} \mathbb{P}(L = l) = e^{\lambda \sigma} \mathbb{E}[L e^{-\mu \sigma L}]. \end{aligned}$$

Proceeding similarly, we obtain

$$\mathbb{P}(R \geq \sigma) \leq \frac{e^{\lambda \sigma}}{\rho} \mathbb{E}[L^2 e^{-\mu \sigma L}].$$

This completes proof of the Theorem 3.3.1.

Now let us assume  $L \sim \text{Binomial}(N, p)$ . Then, by Lemma A.2.1, we have

$$\begin{aligned} \mathbb{E}[L e^{-\mu \sigma L}] &= \frac{N p e^{-\mu \sigma}}{1 - q^N} (p e^{-\mu \sigma} + q)^{N-1}, \\ \mathbb{E}[L^2 e^{-\mu \sigma L}] &= \frac{N p e^{-\mu \sigma}}{1 - q^N} (N p e^{-\mu \sigma} + q) (p e^{-\mu \sigma} + q)^{N-2}. \end{aligned}$$

Therefore, by plugging in  $\theta = \mu - \lambda$ , we get

$$\mathbb{P}(W \geq \sigma) \leq e^{\lambda \sigma} \frac{N p e^{-\mu \sigma}}{1 - q^N} (p e^{-\mu \sigma} + q)^{N-1} = N e^{-\theta \sigma} \left[ \frac{p}{1 - q^N} (p e^{-\mu \sigma} + q)^{N-1} \right],$$

and

$$\begin{aligned} \mathbb{P}(R \geq \sigma) &\leq \frac{e^{\lambda \sigma}}{\rho} \frac{N p e^{-\mu \sigma}}{1 - q^N} (N p e^{-\mu \sigma} + q) (p e^{-\mu \sigma} + q)^{N-2} \\ &= \frac{N e^{-\theta \sigma}}{\rho} \left[ \frac{p}{1 - q^N} (N p e^{-\mu \sigma} + q) (p e^{-\mu \sigma} + q)^{N-2} \right]. \end{aligned}$$

This completes the proof. □

**Lemma A.3.1.** If  $X \sim \text{Pow}(\kappa, \zeta)$  and  $a > 0$ , then

$$\begin{aligned} \mathbb{E}[Xe^{-aX}] &= \frac{\kappa e^{-a} \zeta'(\kappa e^{-a})}{\zeta(\kappa)} \\ \mathbb{E}[X^2 e^{-aX}] &= \frac{\kappa e^{-a}}{\zeta(\kappa)} [\kappa e^{-a} \zeta''(\kappa e^{-a}) + \zeta'(\kappa e^{-a})], \end{aligned}$$

where  $\zeta'$  and  $\zeta''$  are first and second derivatives of  $\zeta$ , respectively.

*Proof of Lemma A.3.1.* See that

$$\mathbb{E}[Xe^{-aX}] = \sum_{l \in \mathbb{N}} l e^{-al} \mathbb{P}(X = l) = \frac{\kappa e^{-a} \zeta'(\kappa e^{-a})}{\zeta(\kappa)}.$$

Now,

$$\mathbb{E}[X(X-1)e^{-aX}] = \sum_{l \in \mathbb{N}} l(l-1) e^{-al} \mathbb{P}(X = l) = \frac{(\kappa e^{-a})^2}{\zeta(\kappa)} \zeta''(\kappa e^{-a}).$$

Therefore, we get

$$\mathbb{E}[X^2 e^{-aX}] = \frac{\kappa e^{-a}}{\zeta(\kappa)} [\kappa e^{-a} \zeta''(\kappa e^{-a}) + \zeta'(\kappa e^{-a})].$$

□

*Proof of Theorem 3.3.2.* First note that  $\alpha(u) := \mathbb{E}[e^{uS}] = \frac{\mu}{\mu-u}$  and  $\beta(u) := \mathbb{E}[e^{-uA_1}] = \frac{\lambda}{\lambda+u}$ , whence we find  $\theta = \mu - \lambda > 0$  such that  $\alpha(\theta)\beta(\theta) = 1$ . Since  $g_l(u) = \alpha(\frac{u}{l^\varphi})$ , the solution to  $g_l(u)\beta(u) = 1$  is given by  $\theta_l := l^\varphi \mu - \lambda > 0$ .

Now consider the scenario conditional on  $\{L = l\}$  for some  $l \in [N]$ . Proceeding in a similar fashion as in the proof of Theorem 3.1.1 and replacing the probabilities and expectations with the corresponding conditional probabilities and expectations respectively, whenever necessary, we get the following bounds on the conditional tail probabilities of the steady state waiting time and the response time as follows

$$\begin{aligned} \mathbb{P}(W \geq \sigma \mid \{L = l\}) &\leq l e^{-\theta_l \sigma}, \\ \mathbb{P}(R \geq \sigma \mid \{L = l\}) &\leq l g_l(\theta_l) e^{-\theta_l \sigma}. \end{aligned}$$

Inserting the value of  $\theta_l$ ,

$$\begin{aligned} \mathbb{P}(W \geq \sigma \mid \{L = l\}) &\leq l e^{\lambda \sigma} \exp(-\mu \sigma l^\varphi), \\ \mathbb{P}(R \geq \sigma \mid \{L = l\}) &\leq \frac{e^{\lambda \sigma}}{\rho} l^2 \exp(-\mu \sigma l^\varphi). \end{aligned}$$

Now, to get bounds on the unconditional probabilities, we utilise the above two upper bounds and note that

$$\begin{aligned} \mathbb{P}(W \geq \sigma) &= \sum_{l \in [N]} \mathbb{P}(W \geq \sigma \mid \{L = l\}) \mathbb{P}(L = l) \\ &\leq e^{\lambda \sigma} \sum_{l \in [N]} l \exp(-\mu \sigma l^\varphi) \mathbb{P}(L = l) = e^{\lambda \sigma} \mathbb{E}[L \exp(-\mu \sigma L^\varphi)]. \end{aligned}$$

Proceeding similarly, we obtain

$$P(R \geq \sigma) \leq \frac{e^{\lambda\sigma}}{\rho} E[L^2 \exp(-\mu\sigma L^\varphi)].$$

This completes proof of Theorem 3.3.2.  $\square$

*Proof of Theorem 3.3.3.* We have  $\alpha_n(u) := E[\exp(uS_n)] = \frac{\mu_n}{\mu_n - u}$  and  $\beta(u) := E[\exp(-uA_1)] = \frac{\lambda}{\lambda + u}$ , whence we find  $\theta_n = \mu_n - \lambda > 0$  such that  $\alpha_n(\theta_n)\beta(\theta_n) = 1$ . Let us denote the conditional MGF of the service times  $S_{n,i}$  at the  $n$ -th server by  $g_l^n$ . Since  $g_l^n(u) = \alpha_n(\frac{u}{l^\varphi})$ , the solution to  $g_l^n(u)\beta(u) = 1$  is given by  $\theta_l^n := l^\varphi \mu_n - \lambda > 0$ . Define

$$\tilde{\theta}_l := \min_{n \in [l]} \theta_l^n.$$

Now consider the scenario conditional on  $\{L = l\}$  for some  $l \in [N]$ . Proceeding in a similar fashion as in the proof of Theorem 3.1.1 and replacing the probabilities and expectations with the corresponding conditional probabilities and expectations respectively, whenever necessary, we get the following bounds on the conditional tail probabilities of the steady state waiting time and the response time as follows

$$\begin{aligned} P(W \geq \sigma \mid \{L = l\}) &\leq l \exp(-\tilde{\theta}_l \sigma), \\ P(R \geq \sigma \mid \{L = l\}) &\leq \left[ \sum_{n \in [l]} g_l^n(\theta_l^n) \right] \exp(-\tilde{\theta}_l \sigma). \end{aligned}$$

Inserting the value of  $\tilde{\theta}_l$  and  $\theta_l^n$ ,

$$\begin{aligned} P(W \geq \sigma \mid \{L = l\}) &\leq l e^{\lambda\sigma} \exp\left(-\min_{n \in [l]} \mu_n \sigma l^\varphi\right), \\ P(R \geq \sigma \mid \{L = l\}) &\leq \frac{e^{\lambda\sigma}}{\lambda} l^\varphi \left( \sum_{n \in [l]} \mu_n \right) \exp\left(-\min_{n \in [l]} \mu_n \sigma l^\varphi\right). \end{aligned}$$

Now, to get bounds on the unconditional probabilities, we utilise the above two upper bounds and note that

$$\begin{aligned} P(W \geq \sigma) &= \sum_{l \in [N]} P(W \geq \sigma \mid \{L = l\}) P(L = l) \\ &\leq e^{\lambda\sigma} \sum_{l \in [N]} l \exp\left(-\min_{n \in [l]} \mu_n \sigma l^\varphi\right) P(L = l) = e^{\lambda\sigma} E[L \exp(-\min_{n \in [L]} \mu_n \sigma L^\varphi)]. \end{aligned}$$

Proceeding similarly, we obtain

$$P(R \geq \sigma) \leq \frac{e^{\lambda\sigma}}{\lambda} E[L^\varphi \left( \sum_{n \in [L]} \mu_n \right) \exp(-\min_{n \in [L]} \mu_n \sigma L^\varphi)].$$

This completes proof of the first part of Theorem 3.3.3.

A SIMPLE CASE Set  $\varphi = 1$ . Now, from  $L \sim \text{Binomial}(N, p)$  we get,

$$\begin{aligned}
& \mathbb{E}[L \exp(-\min_{n \in [L]} \mu_n \sigma L)] \\
&= \mathbb{E}[L \mathbb{E}[\exp(-\min_{n \in [l]} \mu_n \sigma l) \mid L = l]] \\
&= \mathbb{E}[L (\exp(-\sigma L \kappa_1) - (\exp(-\sigma L \kappa_1) - \exp(-\sigma L \kappa_2))(1 - \pi)^L)] \\
&= \frac{N p e^{-\sigma \kappa_1}}{1 - q^N} (p e^{-\sigma \kappa_1} + q)^{N-1} - \frac{N p e^{-(\sigma \kappa_1 - \ln(1-\pi))}}{1 - q^N} (p e^{-(\sigma \kappa_1 - \ln(1-\pi))} + q)^{N-1} \\
&\quad + \frac{N p e^{-(\sigma \kappa_2 - \ln(1-\pi))}}{1 - q^N} (p e^{-(\sigma \kappa_2 - \ln(1-\pi))} + q)^{N-1} \\
&= \frac{N p}{1 - q^N} [e^{-\sigma \kappa_1} (p e^{-\sigma \kappa_1} + q)^{N-1} - (1 - \pi) e^{-\sigma \kappa_1} (p(1 - \pi) e^{-\sigma \kappa_1} + q)^{N-1} \\
&\quad + (1 - \pi) e^{-\sigma \kappa_2} (p(1 - \pi) e^{-\sigma \kappa_2} + q)^{N-1}] \\
&= \frac{N p}{1 - q^N} b_1(\sigma) \left[ 1 - (1 - \pi) \left( \frac{c_1(\sigma) - c_2(\sigma)}{b_1(\sigma)} \right) \right],
\end{aligned}$$

where

$$\begin{aligned}
b_i(\sigma) &:= \exp(-\sigma \kappa_i) (p \exp(-\sigma \kappa_i) + q)^{N-1} \\
c_i(\sigma) &:= \exp(-\sigma \kappa_i) (p(1 - \pi) \exp(-\sigma \kappa_i) + q)^{N-1},
\end{aligned}$$

for  $i = 1, 2$ . Please note that we have used Lemma A.2.1 in the previous derivation. This gives us the bound

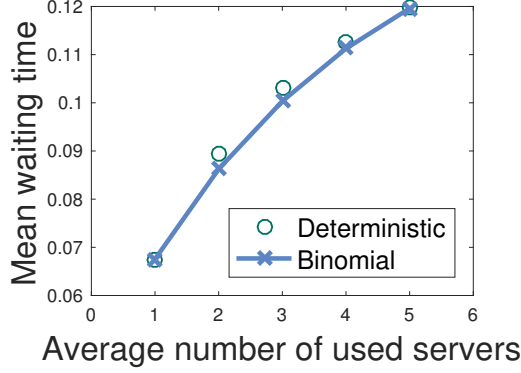
$$\mathbb{P}(W \geq \sigma) \leq e^{\lambda \sigma} \frac{N p}{1 - q^N} b_1(\sigma) \left[ 1 - (1 - \pi) \left( \frac{c_1(\sigma) - c_2(\sigma)}{b_1(\sigma)} \right) \right].$$

This completes the proof.

HIERARCHICAL HYPER-PARAMETER MODEL Set  $\varphi = 1$ . From  $L \sim \text{Binomial}(N, p)$  we get,

$$\begin{aligned}
\mathbb{E}[L \exp(-\min_{n \in [L]} \mu_n \sigma L)] &= \mathbb{E}[L \mathbb{E}[\exp(-\min_{n \in [l]} \mu_n \sigma l) \mid L = l]] \\
&= \mathbb{E}[L \mathbb{E}[\exp(-\sigma l Y_l) \mid L = l]] \\
&= \mathbb{E}[L \frac{L \mu_0}{L \mu_0 + \sigma L} \exp(-\sigma L \lambda)] \\
&= \frac{\mu_0}{\mu_0 + \sigma} \mathbb{E}[L \exp(-\sigma \lambda L)] \\
&= \frac{\mu_0}{\mu_0 + \sigma} \frac{N p e^{-\sigma \lambda}}{1 - q^N} (p e^{-\sigma \lambda} + q)^{N-1}.
\end{aligned}$$





**Figure A.1:** Deterministic versus. stochastic scheduling strategy for an application with specific  $\varphi$  in a heterogeneous FJ system.

Please note that we have used Lemma A.2.1 in the previous derivation. This gives us the bound

$$\begin{aligned}
 P(W \geq \sigma) &\leq e^{\lambda\sigma} \frac{\mu_0}{\mu_0 + \sigma} \frac{Npe^{-\sigma\lambda}}{1 - q^N} \left( pe^{-\sigma\lambda} + q \right)^{N-1} \\
 &= \frac{Np\mu_0}{(1 - q^N)(\mu_0 + \sigma)} \left( pe^{-\sigma\lambda} + q \right)^{N-1}.
 \end{aligned}$$

This completes the proof. □

#### A.4 EVALUATION OF DETERMINISTIC AND STOCHASTIC STRATEGIES

In the following, we compare the average waiting times in a *heterogeneous* FJ system that uses a binomial scheduling strategy with one using a corresponding deterministic strategy. Our aim is to show the benefit of Theorem 3.3.3. We consider renewal job arrivals with exponentially distributed inter-arrival times with parameter  $\lambda = 0.1$  at the ingress of an FJ system with  $N = 5$  servers each of which can be in a *fast* or a *slow* state with probability 0.5. Hence, the service times are exponentially distributed with an average of  $\mu = 1$  in the first state, and  $\mu = 0.5$  in the second. We assume an application with a weak parallelisation benefit  $\varphi = 0.2$ . The rationale here is to let the system switch between a regime where the synchronisation cost outweighs the scaling benefit, and another regime where the opposite holds true. Given a pool of  $N$  available servers, Figure A.1 compares the mean waiting time under a deterministic strategy that uses  $1 \leq L' \leq N$  servers to a stochastic strategy that uses an average number of servers  $E[L] = L'$ . This example shows that the stochastic strategy is superior to a comparable deterministic one. This strengthens our argument that for a known application that runs on a given FJ system, Theorem 3.3.3 provides the optimal scheduling strategy.



## SUPPLEMENTARY MATERIAL TO CHAPTER 4

### B.1 LARGE DEVIATIONS PRINCIPLE

*Proof of Theorem 4.1.1.* In the light of A1, A2, A3, and A4, the following statements are immediate from known results on large deviations of Markov additive process (Iscoe, Ney, and Nummelin 1985; Ney and Nummelin 1987),

**B1** For all  $\theta \in \mathcal{D}_0$ , the transformed kernel  $\tilde{L}$  in (4.1.5) has a maximal, real, simple eigenvalue  $\lambda(\theta)$ .

**B2** The corresponding right eigenfunction  $\{r(c, \theta); c \in \mathbb{E}\}$  satisfying

$$\lambda(\theta)r(c, \theta) = \int_{\mathbb{R}} \tilde{L}(c, d\tau; \theta)r(\tau, \theta),$$

is positive and bounded above.

**B3**  $\mathcal{D}\lambda = \mathcal{D}\tilde{\nu} = \mathcal{D}_0$ .

**B4** Define the filtration

$$\mathcal{F}_k := \sigma(\{(C_i, Q_i)\}_{i \in [k]}), \quad (2.1.1)$$

the  $\sigma$ -algebra generated by the history of the process  $\{(C_i, Q_i)\}_{i \in [k]}$  till and including time point  $k$ . Define

$$M_k(\theta) := \exp(aQ_k - k\Lambda(\theta))r(C_k, \theta), \quad (2.1.2)$$

where  $\Lambda(\theta) := \log \lambda(\theta)$ . The process  $M_k(\theta)$  is a martingale with respect to the filtration  $\mathcal{F}_k$ .

**B5**  $\lambda(\theta) \rightarrow \infty$  as  $\theta \rightarrow \text{Bnd } \mathcal{D}$  or  $\|\theta\| \rightarrow \infty$ . This further implies essential smoothness of  $\Lambda$ . This is important for the application of Ellis' theorem to establish an LDP.

Note that B1 and B2 are generalisations of the well known *Perron-Frobenius* theorem for real matrices with positive entries. However, when the state space  $\mathbb{E}$  is not finite, one could still obtain similar results. The existence, and properties B1 and B2 follow from Harris (1963) and Iscoe, Ney, and Nummelin (1985). Define  $\pi : \mathcal{E} \rightarrow [0, 1]$  to be the invariant probability measure for  $L$  defined in (4.1.3). The following large deviations principle holds (Iscoe, Ney, and Nummelin 1985) for the sequence of probability measures  $\{L^k(x, F \times \cdot)\}_{k \in \mathbb{N}_0}$  on  $(\mathbb{R}^N, \mathcal{B}(\mathbb{R}^N))$ ,

$$\limsup_{k \rightarrow \infty} k^{-1} \log L^k(x, F \times kG) \leq - \inf_{y \in \text{Cl } G} \Lambda^*(y), \quad (2.1.3)$$

$$\liminf_{k \rightarrow \infty} k^{-1} \log L^k(x, F \times kG) \geq - \inf_{y \in \text{Int } G} \Lambda^*(y), \quad (2.1.4)$$

for  $x \in \mathbb{E}, F \in \mathcal{E}, G \in \mathcal{B}(\mathbb{R}^N)$ , where  $\Lambda^*(y) := \sup_{z \in \mathbb{R}^N} \{zy - \Lambda(z)\}$  and  $F$  is such that  $\pi(F) > 0$ .

In order to derive a large deviations principle for the waiting times for our queueing system defined in (4.1.2), consider the following map  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  defined as

$$f(s) := \max\{s_1, s_2, \dots, s_N\}, \quad (2.1.5)$$

where  $s := (s_1, s_2, \dots, s_N) \in \mathbb{R}^N$ . Note that  $f$  is a continuous map on  $\mathbb{R}^N$  with respect to the topology endowed by the Borel open sets. Therefore, by the contraction principle for continuous maps (Dembo and Zeitouni 2010, Theorem 4.2.1),  $\{f(Q_k)\}_{k \in \mathbb{N}_0}$  satisfies a large deviations principle with good rate function

$$J(y) := \inf_{x \in f^{-1}(y)} \Lambda^*(x) = \inf_{x \in Y_N(y)} \Lambda^*(x), \quad (2.1.6)$$

where  $Y_N$  is defined in (4.1.6). Notice that  $f(Q_k)$  is simply  $W_k := \max(X_{1,k}, X_{2,k}, \dots, X_{N,k})$  with  $W \stackrel{\mathcal{D}}{=} \max_{k \in \mathbb{N}_0} W_k$ . Therefore, by virtue of the contraction principle, we get

$$\begin{aligned} \limsup_{k \rightarrow \infty} k^{-1} \log \mathbb{P}(W_k \in B) &\leq - \inf_{y \in \text{Cl } B} J(y) \\ \liminf_{k \rightarrow \infty} k^{-1} \log \mathbb{P}(W_k \in B) &\geq - \inf_{y \in \text{Int } B} J(y), \end{aligned}$$

for all  $B \in \mathcal{B}(\mathbb{R})$ . This completes the proof. □

## B.2 FURTHER DERIVATIONS

*Derivation of (4.1.14).* We wish to solve the following integral equation for  $\lambda^{(n)}$ , and  $r_n$ ,

$$\int_a^b \exp\left(-\frac{|y-x|}{\sigma}\right) r_n(x, s) dx = U_n(y, s) \exp(\lambda^{(n)}(s)) r_n(y, s),$$

where  $U_n(y, s) = \left(1 + \frac{s}{y}\right) \left(1 - \frac{s}{\mu^{(n)}}\right) u(y)$  and  $u(y) = \int_a^b \exp\left(-\frac{|x-y|}{\sigma}\right) dx$ . Our strategy is to differentiate the above integral equation with respect to  $y$  twice and then get a nonlinear ODE, which can be solved numerically. Therefore, separating the integral into two parts we get

$$\begin{aligned} \exp\left(-\frac{y}{\sigma}\right) \int_a^y \exp\left(\frac{x}{\sigma}\right) r_n(x, s) dx + \exp\left(\frac{y}{\sigma}\right) \int_y^b \exp\left(-\frac{x}{\sigma}\right) r_n(x, s) dx \\ = U_n(y, s) \exp(\lambda^{(n)}(s)) r_n(y, s). \end{aligned}$$

Differentiating once with respect to  $y$ , we get

$$\begin{aligned} -\frac{1}{\sigma} \exp\left(-\frac{y}{\sigma}\right) \int_a^y \exp\left(\frac{x}{\sigma}\right) r_n(x, s) dx + \frac{1}{\sigma} \exp\left(\frac{y}{\sigma}\right) \int_a^y \exp\left(-\frac{x}{\sigma}\right) r_n(x, s) dx \\ = U'_n(y, s) \exp(\lambda^{(n)}(s)) r_n(y, s) + U_n(y, s) \exp(\lambda^{(n)}(s)) r'_n(y, s). \end{aligned}$$

Differentiating once again with respect to  $y$ , we get

$$\begin{aligned} & \frac{1}{\sigma^2} \left( \exp\left(-\frac{y}{\sigma}\right) \int_a^y \exp\left(\frac{x}{\sigma}\right) r_n(x, s) dx + \exp\left(\frac{y}{\sigma}\right) \int_y^b \exp\left(-\frac{x}{\sigma}\right) r_n(x, s) dx - 2\sigma r_n(y, s) \right) \\ &= U_n''(y, s) \exp(\lambda^{(n)}(s)) r_n(y, s) + 2U_n'(y, s) \exp(\lambda^{(n)}(s)) r_n'(y, s) \\ &+ U_n(y, s) \exp(\lambda^{(n)}(s)) r_n''(y, s). \end{aligned}$$

Since the left hand side is  $\frac{1}{\sigma^2} U_n(y, s) \exp(\lambda^{(n)}(s)) r_n(y, s)$ , after rearrangement of terms, we get

$$r_n''(y, s) + 2 \frac{U_n'(y, s)}{U_n(y, s)} r_n'(y, s) + \left( \frac{U_n''(y, s)}{U_n(y, s)} - \frac{1}{\sigma^2} \left( 1 - \frac{2\sigma \exp(-\lambda^{(n)}(s))}{U_n(y, s)} \right) \right) r_n(y, s) = 0.$$

□

*Proof of Theorem 4.1.2.* In the light of A1, A2, A3, and A4, the following statements are immediate from known results in probability theory, such as Iscoe, Ney, and Nummelin (1985) and Ney and Nummelin (1987),

**C1** For all  $n \in [N]$  and  $\theta \in \mathcal{D}\lambda^{(n)}$ ,  $\exp(\lambda^{(n)}(\theta))$  is the simple maximal eigenvalue of  $\tilde{K}_n$ .

**C2** The corresponding right eigenfunction  $\{r_n(c, \theta); c \in \mathbb{E}\}$  satisfying

$$\exp(\lambda^{(n)}(\theta)) r_n(c, \theta) = \int_{\mathbb{R}} \tilde{K}_n(c, d\tau; \theta) r_n(\tau, \theta),$$

is positive and bounded above.

**C3** For all  $n \in [N]$ , the functions  $\lambda^{(n)}$  and  $\lambda_k^{(n)}$ ,  $k \in \mathbb{N}$  are both strictly convex and essentially smooth.

**C4** Recall the filtration  $\mathcal{F}_k$  defined in (2.1.1). For each  $n \in [N]$ , define

$$M_k^{(n)}(s) := \exp(sX_{n,k} - k\lambda^{(n)}(s)) r_n(C_k, s). \quad (2.2.1)$$

Then,  $M_k^{(n)}(s)$  is a martingale with respect to the filtration  $\mathcal{F}_k$ .

The existence, and properties C1 and C2 follow from Harris (1963) and Iscoe, Ney, and Nummelin (1985). The statements C3 and C4 are proved in Iscoe, Ney, and Nummelin (1985). Also, see Duffield (1994). In the following, we normalise  $r_n(\cdot, \theta)$  so that  $E[r_n(C_0, \theta)] = 1$ , for each  $n \in [N]$ .

Having constructed the martingales  $M_k^{(n)}(s)$ , we can apply Doob's maximal inequality (Durrett 2010a) to obtain

$$P(\max_{k \in \mathbb{N}_0} X_{n,k} \geq w) \leq \phi_n(s) \exp(-sw), \quad (2.2.2)$$

for all  $s \in \mathcal{D}\lambda^{(n)}$ , following Theorem 3 of Duffield (1994). In particular, we get

$$P(\max_{k \in \mathbb{N}_0} X_{n,k} \geq w) \leq \phi_n(\theta_n) \exp(-\theta_n w), \quad (2.2.3)$$

where  $\theta_n := \sup\{s > 0 \mid \lambda^{(n)}(s) \leq 0\}$  and  $\phi_n(s) := \text{ess sup}\{\mathbb{1}(X_{n,1} > 0)/r_n(C_1, s)\}$ , after having normalised  $r_n(\cdot, \theta)$  so that  $E[r_n(C_0, \theta)] = 1$ , for each  $n \in [N]$ . The final bound is obtained as follows

$$\begin{aligned} P(W \geq w) &= P(\max_{k \in \mathbb{N}_0} \max_{n \in [N]} X_{n,k} \geq w) = P(\max_{n \in [N]} \max_{k \in \mathbb{N}_0} X_{n,k} \geq w) \\ &\leq \sum_{n \in [N]} P(\max_{k \in \mathbb{N}_0} X_{n,k} \geq w) \leq \sum_{n \in [N]} \phi_n(\theta_n) \exp(-\theta_n w). \end{aligned}$$

This completes the proof.  $\square$

**Remark B.2.1.** For the computation of the MGF for the blocking system, we make use of the following statistical result. Consider a finite collection of independent random variables  $\{U_n\}_{n \in [N]}$  such that  $U_n$  is exponentially distributed with rate  $\mu_n$ , for each  $n \in [N]$ . Write  $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ . Then, the MGF of  $V := \max_{n \in [N]} U_n$  is given by

$$E[\exp(sV)] = \beta(\mu; s) := \sum_{S \in \{A \subset [N] \mid A \neq \emptyset\}} (-1)^{|S|+1} \frac{(\sum_{i \in S} \mu_i)}{(\sum_{i \in S} \mu_i) - s}. \quad (2.2.4)$$

*Proof of Result B.2.1.* The CDF of  $Z$  is given by  $P(V \leq z) = \prod_{i \in [N]} (1 - \exp(-\mu_i z))$ , whence we derive the Probability Density Function (PDF) of  $Z$  as

$$\begin{aligned} f_V(z) &= \sum_{j \in [N]} \mu_j \exp(-\mu_j z) \left[ \prod_{i \in [N] \setminus \{j\}} (1 - \exp(-\mu_i z)) \right] \\ &= \sum_{j \in [N]} \mu_j \exp(-\mu_j z) \left[ 1 + \sum_{S \in \{A \subset [N] \setminus \{j\} \mid A \neq \emptyset\}} (-1)^{|S|} \prod_{i \in S} \exp(-\mu_i z) \right] \\ &= \sum_{j \in [N]} \mu_j \exp(-\mu_j z) \left[ \sum_{S \in \{A \subset [N] \setminus \{j\}\}} (-1)^{|S|} \exp(-z \sum_{i \in S} \mu_i) \right] \\ &= \sum_{j \in [N]} \mu_j \sum_{S \in \{A \subset [N] \setminus \{j\}\}} (-1)^{|S|} \exp(-z \sum_{i \in S \cup \{j\}} \mu_i) \\ &= \sum_{j \in [N]} \mu_j \sum_{S \in \{A \subset [N] \mid j \in A\}} (-1)^{|S|+1} \exp(-z \sum_{i \in S} \mu_i) \\ &= \sum_{S \in \{A \subset [N] \mid A \neq \emptyset\}} (-1)^{|S|+1} (\sum_{i \in S} \mu_i) \exp(-z \sum_{i \in S} \mu_i). \end{aligned}$$

Therefore, the MGF of  $Z$  is given by

$$E[\exp(\theta V)] = \int_0^\infty \exp(\theta z) f_V(z) dz = \sum_{S \in \{A \subset [N] \mid A \neq \emptyset\}} (-1)^{|S|+1} \frac{(\sum_{i \in S} \mu_i)}{(\sum_{i \in S} \mu_i) - \theta}.$$

This completes the proof.  $\square$

## SUPPLEMENTARY MATERIAL TO CHAPTER 5

## C.1 STATISTICAL RESULTS

## C.1.1 Moments of order statistics

Let  $X_1, X_2, \dots, X_N$  be independent positive-valued random variables with *absolutely continuous* CDFs  $F_1, F_2, \dots, F_N$ . Let the corresponding order statistics be  $Y_1 \leq Y_2 \leq \dots \leq Y_N$ . Write  $F := (F_1, F_2, \dots, F_N)^T$  and  $1 - F := (1 - F_1, 1 - F_2, \dots, 1 - F_N)^T$ . The distribution of the  $r$ -th order statistic can be elegantly written in terms of certain permanents as (Bapat and Beg 1989, Theorem 4.1),

$$P(Y_r \leq y) = \sum_{i=r}^N \frac{1}{i!(N-i)!} \text{per} \left[ \frac{F(y)}{i} \frac{1-F(y)}{N-i} \right], \quad (3.1.1)$$

where  $\left[ \frac{F(y)}{i} \frac{1-F(y)}{N-i} \right]$  denotes the matrix whose first  $i$  columns are  $F(y)$  and the last  $N-i$  columns are  $1 - F(y)$ ,  $\text{per } A := \sum_{\sigma \in \Theta(N)} \prod_{i=1}^N a_{i, \sigma(i)}$  denotes the permanent of an  $N \times N$  real matrix  $A = ((a_{i,j}))_{i,j \in [N]}$ , and  $\Theta(N)$  denote the class of all permutations of  $[N]$ . Using (3.1.1), we derive the expected values of the order statistics (Bapat and Beg 1989; Barakat and Abdelkader 2004).

**Remark C.1.1.** For  $r \in [N]$ , the mean of  $Y_r$  can be conveniently written in terms of  $\mu$ -operators given by

$$E[Y_r] = \mu_r F := \sum_{j=N-r+1}^N (-1)^{j-(N-r-1)} \binom{j-1}{N-r} \mathbb{M}_j F,$$

where the  $\mathbb{M}_j$ -operators, for  $j \in [N]$ , are defined as

$$\mathbb{M}_j F := \sum_{S \in \{A \subseteq [N] : |A|=j\}} \int_0^\infty \left( \prod_{i \in S} (1 - F_i(x)) \right) dx. \quad (3.1.2)$$

*Proof of Remark C.1.1.* The proof follows from Bapat and Beg (1989) and Barakat and Abdelkader (2004). However, for the sake of completeness, we furnish a brief sketch here. Define  $H_r(y) := P(Y_r \leq y)$  for  $r \in [N]$ , where  $P(Y_r \leq y)$  is given in (3.1.1). Then, the mean can be obtained by performing the following integral

$$E[Y_r] = \int_0^\infty (1 - H_r(y)) dy.$$

Observe that, we can derive the following recursion relation from (3.1.1), for  $r \geq 2$ ,

$$H_{r-1}(y) = H_r(y) + \frac{1}{(r-1)!(N-r+1)!} \text{per} \left[ \frac{F(y)}{r-1} \frac{1-F(y)}{N-r+1} \right], \quad (3.1.3)$$

where the permanent of a real  $N \times N$  matrix  $A := ((a_{i,j}))_{i,j \in [N]}$  is given by

$$\text{per } A := \sum_{\sigma \in \Theta(N)} \prod_{i=1}^N a_{i,\sigma(i)},$$

and  $\Theta(N)$  denote the class of all permutations of  $[N]$ . Plugging in the definition of the permanent, we rewrite (3.1.3) as

$$H_{r-1}(y) = H_r(y) + \frac{1}{(r-1)!(N-r+1)!} \sum_{\sigma \in \Theta(N)} \prod_{i=1}^N a_{i,\sigma(i)}(y),$$

where

$$a_{i,\sigma(i)}(y) = \begin{cases} F_i(y) & \text{if } 1 \leq \sigma(i) \leq r-1, \\ 1 - F_i(y) & \text{if } r \leq \sigma(i) \leq N. \end{cases} \quad (3.1.4)$$

Rearranging the terms in the recurrence relation, we get

$$1 - H_r(y) = 1 - H_{r-1}(y) + \frac{1}{(r-1)!(N-r+1)!} \sum_{\sigma \in \Theta(N)} \prod_{i=1}^N a_{i,\sigma(i)}(y).$$

Integrating both sides and using the  $\mu$ -operators, we get

$$\mu_r F = \mu_{r-1} F + \frac{1}{(r-1)!(N-r+1)!} \sum_{\sigma \in \Theta(N)} \int_0^\infty \prod_{i=1}^N a_{i,\sigma(i)}(y) \, dy = \mu_{r-1} F + K_r F,$$

where the operator  $K_r$  is given by

$$K_r F := \frac{1}{(r-1)!(N-r+1)!} \sum_{\sigma \in \Theta(N)} \int_0^\infty \prod_{i=1}^N a_{i,\sigma(i)}(y) \, dy.$$

Note that there are  $r-1$  terms involving  $F_i(y)$  and  $N-r+1$  terms involving  $1-F_i(y)$  in the product, for each permutation  $\sigma \in \Theta(N)$ . Therefore, we have

$$K_r F = \sum_{S \in \{A \subseteq [N] : |A|=r-1\}} \int_0^\infty \left( \prod_{j \in S} F_j(y) \right) \left( \prod_{j \in S^c} (1 - F_j(y)) \right) \, dy.$$

Let us rewrite  $K_r$ -operators in the following way to get an identity

$$K_r F \equiv \sum_{j=1}^r (-1)^{j-1} c(j, r, N) \mathbb{M}_{N-r+j} F, \quad (3.1.5)$$



where  $c(j, r, N)$ 's are suitable counting coefficients so that the above identity holds true with  $\mathbb{M}$ -operators defined by

$$\mathbb{M}_j F := \sum_{S \in \{A \subseteq [N] : |A|=j\}} \int_0^\infty \left( \prod_{i \in S} (1 - F_i(x)) \right) dx.$$

Notice that the number of terms under the summation over  $S \subseteq [N]$  with  $|S| = r - 1$  is  $\binom{N}{r-1}$ , while that under the summation over  $S \subseteq [N]$  with  $|S| = N - r + j$  appearing in the computation of  $\mathbb{M}_{N-r+j} F$  is  $\binom{N}{N-r+j}$ . Therefore, by applying multiplication principle of combinatorial analysis, the counting coefficients  $c(j, r, N)$  must satisfy

$$\binom{N}{r-1} \binom{r-1}{j-1} = c(j, r, N) \binom{N}{N-r+j},$$

in order for the above identity in (3.1.5) to hold true (see Barakat and Abdelkader (2004)). Therefore, we get

$$c(j, r, N) = \binom{N-r+j}{j-1}, \quad (3.1.6)$$

and we get the following recursion relation, for  $2 \leq r \leq N$ ,

$$\mu_r F = \mu_{r-1} F + \sum_{j=1}^r (-1)^{j-1} \binom{N-r+j}{j-1} \mathbb{M}_{N-r+j} F. \quad (3.1.7)$$

Observe that  $\mu_1 F = \mathbb{M}_N F$  and  $\mu_2 F = \mathbb{M}_{N-1} F - (N-1)\mathbb{M}_N F$ . Thereby from (3.1.7), the claim

$$\mu_r F = \sum_{j=N-r+1}^N (-1)^{j-(N-r+1)} \binom{j-1}{N-r} \mathbb{M}_j F \quad (3.1.8)$$

follows by induction on  $r$ . The induction is proved in Barakat and Abdelkader (2004) and we do not repeat it here. This completes the proof.  $\square$

## C.2 ADDITIONAL DERIVATIONS

*Derivation of  $\psi(k_1, k_2) \geq \psi(k_1 + 1, k_2 - 1) \iff \frac{I_p(k_1, k_2)}{I_{1-p}(k_2-1, k_1+1)} \geq \frac{\lambda_2}{\lambda_1}$ . Write  $p := \frac{\lambda_1}{\lambda_1 + \lambda_2}$  and  $q := \frac{\lambda_2}{\lambda_1 + \lambda_2}$ . Then,*

$$\psi(k_1, k_2) = \frac{k_1}{\lambda_1} + \frac{K - k_1}{\lambda_2} - \frac{1}{\lambda_1 + \lambda_2} \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{K-k_1-1} \binom{n_1 + n_2}{n_1} p^{n_1} q^{n_2}.$$

Now,

$$\begin{aligned} \psi(k_1, k_2) - \psi(k_1 + 1, k_2 - 1) &= \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) - \frac{1}{\lambda_1 + \lambda_2} \left[ \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{K-k_1-1} \binom{n_1 + n_2}{n_1} p^{n_1} q^{n_2} \right. \\ &\quad \left. - \sum_{n_1=0}^{k_1} \sum_{n_2=0}^{K-k_1-2} \binom{n_1 + n_2}{n_1} p^{n_1} q^{n_2} \right]. \end{aligned}$$

Simplifying further, we get

$$\begin{aligned}
& \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{K-k_1-1} \binom{n_1+n_2}{n_1} p^{n_1} q^{n_2} - \sum_{n_1=0}^{k_1} \sum_{n_2=0}^{K-k_1-2} \binom{n_1+n_2}{n_1} p^{n_1} q^{n_2} \\
&= \sum_{n_1=0}^{k_1-1} \left[ \sum_{n_2=0}^{K-k_1-2} \binom{n_1+n_2}{n_1} p^{n_1} q^{n_2} + \binom{n_1+K-k_1-1}{n_1} p^{n_1} q^{K-k_1-1} \right] \\
&\quad - \left[ \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{K-k_1-2} \binom{n_1+n_2}{n_1} p^{n_1} q^{n_2} + \sum_{n_2=0}^{K-k_1-2} \binom{k_1+n_2}{k_1} p^{k_1} q^{n_2} \right] \\
&= \sum_{n_1=0}^{k_1-1} \binom{n_1+K-k_1-1}{n_1} p^{n_1} q^{K-k_1-1} - \sum_{n_2=0}^{K-k_1-2} \binom{k_1+n_2}{k_1} p^{k_1} q^{n_2} \\
&= \frac{1}{q} \sum_{n_1=0}^{k_1-1} \binom{n_1+K-k_1-1}{n_1} p^{n_1} q^{K-k_1} - \frac{1}{p} \sum_{n_2=0}^{K-k_1-2} \binom{k_1+n_2}{k_1} p^{k_1+1} q^{n_2} \\
&= \frac{1}{q} F(k_1-1; K-k_1, p) - \frac{1}{p} F(K-k_1-2; k_1+1, p) \\
&= \frac{1}{q} I_q(K-k_1, k_1) - \frac{1}{p} I_p(k_1+1, K-k_1-1),
\end{aligned}$$

where  $F(.; n, s)$  is the CDF of a negative binomial distribution with parameters  $n$  and  $s$  (denoted as NB( $n, s$ )) and  $I_x(a, b)$  is the regularised  $\beta$ -function given by

$$I_x(a, b) := \frac{\int_0^x t^{a-1} (1-t)^{b-1} dt}{\int_0^1 t^{a-1} (1-t)^{b-1} dt}.$$

Therefore, we have

$$\begin{aligned}
& \psi(k_1, k_2) - \psi(k_1+1, k_2-1) \\
&= \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) - \frac{1}{\lambda_1 + \lambda_2} \left[ \frac{1}{q} I_q(K-k_1, k_1) - \frac{1}{p} I_p(k_1+1, K-k_1-1) \right] \\
&= \frac{1}{\lambda_2} I_p(k_1, K-k_1) - \frac{1}{\lambda_1} I_q(K-k_1-1, k_1+1),
\end{aligned}$$

whence we get

$$\begin{aligned}
& \psi(k_1, k_2) \geq \psi(k_1+1, k_2-1) \\
&\iff \frac{1}{\lambda_2} I_p(k_1, K-k_1) \geq \frac{1}{\lambda_1} I_q(K-k_1-1, k_1+1) \\
&\iff \frac{I_p(k_1, K-k_1)}{I_q(K-k_1-1, k_1+1)} \geq \frac{\lambda_2}{\lambda_1}.
\end{aligned}$$

This completes the proof. The above allows finding the optimal allocation  $k_{\text{opt}}$  in an iterative fashion. Given  $\lambda_1 < \lambda_2$ , and we sequentially check  $(0, K), (1, K-1), (2, K-2), \dots$  and so on as long as the ratio of the two regularised  $\beta$ -functions is greater than  $\lambda_2/\lambda_1$ . The objective function  $\psi$  is monotonically decreasing in its first argument  $k_1$

in this range. The optimal choice is the last allocation in this sequence when the ratio of the regularised  $\beta$ -functions is greater than or equal to  $\lambda_2/\lambda_1$ , beyond this point  $\psi$  is again monotonically increasing in its first argument  $k_1$ . See Figure 5.1. If  $\lambda_1 > \lambda_2$ , we interchange (relabel) the paths and proceed as before. Since the optimal allocation is  $(K/2, K/2)$  (or the nearest integers depending on whether  $K$  is even or odd) when  $\lambda_1 = \lambda_2$ . □

*Proof of optimality when  $\lambda_1 = \lambda_2$ .* Suppose  $\lambda_1 = \lambda_2 = \lambda$ ,

$$\psi(k_1, k_2) = \frac{K}{\lambda} - \frac{1}{2\lambda} \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{k_2-1} \binom{n_1+n_2}{n_1} \left(\frac{1}{2}\right)^{n_1+n_2}.$$

For natural numbers  $c, m$  with  $c > m$ , define

$$G(m) = \sum_{n_1=0}^m \sum_{n_2=0}^{c-m} \binom{n_1+n_2}{n_1} \left(\frac{1}{2}\right)^{n_1+n_2}.$$

Then

$$G(m+1) - G(m) = \sum_{n_2=0}^{c-m-1} \binom{m+1+n_2}{m+1} \left(\frac{1}{2}\right)^{m+1+n_2} - \sum_{n_2=0}^m \binom{c-m+n_2}{c-m} \left(\frac{1}{2}\right)^{c-m+n_2}.$$

Comparing the last summands under two summations,

$$\binom{c}{m+1} \left(\frac{1}{2}\right)^c \geq \binom{c}{c-m} \left(\frac{1}{2}\right)^c \iff \frac{c-m}{m+1} \geq 1 \iff 2m \leq c-1.$$

Further,

$$\frac{c-m}{m+1} \geq 1 \implies \binom{c-i}{m+1} \left(\frac{1}{2}\right)^{c-i} \geq \binom{c-i}{c-m} \left(\frac{1}{2}\right)^{c-i}.$$

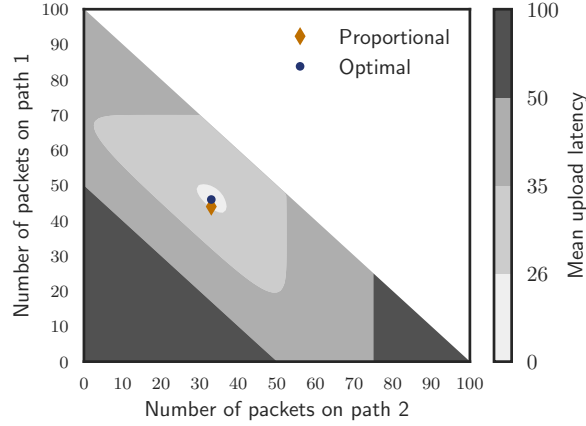
Summing over  $i = 0, 1, \dots, \min(c-m-1, m)$ , we get

$$2m \leq c-1 \implies G(m+1) \geq G(m).$$

The converse can be proved using similar arguments. Therefore,  $G(m)$  attains maxima at  $\lfloor \frac{c+1}{2} \rfloor$ . Consequently,  $\psi$  is minimum when  $n_1 - 1 = \lfloor \frac{k_1-1+k_2-1+1}{2} \rfloor$  i.e.,  $n_1 = \lfloor \frac{K+1}{2} \rfloor$ . This completes the proof. □

*Derivation of N-path case with exponential delays.* Suppose the  $i$ -th path has an exponential delay with rate  $\lambda_i$  (i.e.,  $D_{i,j}$ 's follow an exponential distribution with mean  $1/\lambda_i$ , for  $i \in [N]$ ). Let  $\mathbf{k} = (k_1, k_2, \dots, k_N) \in \Lambda(N, K)$  be our allocation. Then, the end-to-end delay can be expressed as  $D := \max(D_1^{(k_1)}, D_2^{(k_2)}, \dots, D_N^{(k_N)})$  where  $D_i^{(k_i)} := \sum_{j=1}^{k_i} D_{i,j}$ . Note that  $D_i^{(k_i)}$  follows a gamma distribution with parameters  $k_i$  and  $\lambda_i$  (which is the same as an Erlang distribution in this case). Therefore, the CDF  $F_i^{(k_i)}$  of  $D_i^{(k_i)}$  is given by

$$F_i^{(k_i)}(x) = 1 - \sum_{m=0}^{k_i-1} e^{-\lambda_i x} \frac{(\lambda_i x)^m}{m!} \quad \text{for } i \in [N]. \quad (3.2.1)$$



**Figure C.1:** Three heterogeneous paths with exponential delays with rates 2, 1.5 and 1 are considered. The optimal allocation is centred at the innermost contour. The mean delay is high if the stronger paths (the first two) are grossly under-utilised (see bottom left corner). The proportional allocation is expectedly close to the optimal one. The number of packets considered in this example is 100.

Stacking into a column vector  $\mathbf{F}^{(k)} := (F_1^{(k_1)}, F_2^{(k_2)}, \dots, F_N^{(k_N)})$ , and following Remark C.1.1, we find explicitly

$$\psi(\mathbf{k}) = \mu_n \mathbf{F}^{(k)} = \sum_{j=1}^n (-1)^{j+1} \mathbb{M}_j \mathbf{F}^{(k)}, \quad (3.2.2)$$

where  $\mathbb{M}_j$ 's are as defined in (3.1.2) of Remark C.1.1. Please note that

$$\mathbb{M}_j \mathbf{F}^{(k)} = \sum_{S \in \{A \subseteq [N] : |A|=j\}} \int_0^\infty \left[ \prod_{i \in S} \sum_{m_i=0}^{k_i-1} e^{-\lambda_i x} \frac{(\lambda_i x)^{m_i}}{m_i!} \right] dx.$$

Using  $\int_0^\infty e^{ax} x^{b-1} dx = \frac{\Gamma(b)}{a^b}$  and rearranging terms, we get

$$\psi(\mathbf{k}) = \sum_{S \in \{A \subseteq [N] : A \neq \emptyset\}} (-1)^{|S|+1} \left[ \sum_{n_i \in [k_i-1] \cup \{0\} : i \in S} \left( \prod_{i \in S} \frac{\lambda_i^{n_i}}{n_i!} \right) \frac{\Gamma(\sum_{i \in S} n_i + 1)}{(\sum_{i \in S} \lambda_i)^{\sum_{i \in S} n_i + 1}} \right].$$

This completes the derivation. Please see Bapat and Beg (1989) and Barakat and Abdelkader (2004) for more on this and other similar examples.  $\square$

**Example C.2.1.** When there are three paths admitting exponential delays with parameters  $\lambda_1, \lambda_2$  and  $\lambda_3$  respectively, the expression for mean delay corresponding to an allocation  $\mathbf{k} = (k_1, k_2, k_3) \in \Lambda(3, K)$  simplifies to

$$\begin{aligned} \psi(\mathbf{k}) = & \frac{k_1}{\lambda_1} + \frac{k_2}{\lambda_2} + \frac{k_3}{\lambda_3} - \frac{1}{\lambda_1 + \lambda_2} \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{k_2-1} \frac{(n_1 + n_2)!}{n_1! n_2!} \left(\frac{\lambda_1}{\lambda_1 + \lambda_2}\right)^{n_1} \left(\frac{\lambda_2}{\lambda_1 + \lambda_2}\right)^{n_2} \\ & - \frac{1}{\lambda_2 + \lambda_3} \sum_{n_2=0}^{k_2-1} \sum_{n_3=0}^{k_3-1} \frac{(n_2 + n_3)!}{n_2! n_3!} \left(\frac{\lambda_2}{\lambda_2 + \lambda_3}\right)^{n_2} \left(\frac{\lambda_3}{\lambda_2 + \lambda_3}\right)^{n_3} \\ & - \frac{1}{\lambda_3 + \lambda_1} \sum_{n_3=0}^{k_3-1} \sum_{n_1=0}^{k_1-1} \frac{(n_3 + n_1)!}{n_3! n_1!} \left(\frac{\lambda_3}{\lambda_3 + \lambda_1}\right)^{n_3} \left(\frac{\lambda_1}{\lambda_3 + \lambda_1}\right)^{n_1} \\ & + \frac{1}{\lambda_1 + \lambda_2 + \lambda_3} \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{k_2-1} \sum_{n_3=0}^{k_3-1} \frac{(n_1 + n_2 + n_3)!}{n_1! n_2! n_3!} \left(\frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3}\right)^{n_1} \\ & \times \left(\frac{\lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}\right)^{n_2} \left(\frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}\right)^{n_3}. \end{aligned} \quad (3.2.3)$$

The expression of the mean delay above can be minimised to find the optimal allocation. In Figure C.1, we consider three heterogeneous paths with exponential delays with rates  $\lambda_i$  equal to  $\{2, 1.5, 1\}$  respectively. The near-optimality of the proportional allocation is observed here too (centred at the innermost contour).

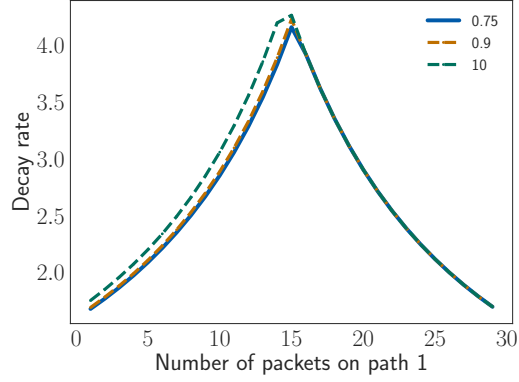
*Derivations for Example 5.2.1.* Write  $a_i = \lambda_i / (\lambda_1 + \lambda_2 + \lambda_3)$  for  $i = 1, 2, 3$ . Then,

$$\begin{aligned} \eta_1(\mathbf{k}) = & \int_0^\infty \left( \sum_{n_1=0}^{k_1-1} e^{-\lambda_1 x} \frac{(\lambda_1 x)^{n_1}}{n_1!} \right) \left( \sum_{n_2=0}^{k_2-1} e^{-\lambda_2 x} \frac{(\lambda_2 x)^{n_2}}{n_2!} \right) \left( \sum_{n_3=0}^{k_3-1} e^{-\lambda_3 x} \frac{(\lambda_3 x)^{n_3}}{n_3!} \right) dx \\ = & \frac{1}{\lambda_1 + \lambda_2 + \lambda_3} \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{k_2-1} \sum_{n_3=0}^{k_3-1} \frac{(n_1 + n_2 + n_3)!}{n_1! n_2! n_3!} a_1^{n_1} a_2^{n_2} a_3^{n_3}. \end{aligned} \quad (3.2.4)$$

Similarly,

$$\begin{aligned} \eta_2(\mathbf{k}) = & \mathbb{M}_2 F^{(\mathbf{k})} - 2\mathbb{M}_3 F^{(\mathbf{k})} \\ = & \frac{1}{\lambda_1 + \lambda_2} \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{k_2-1} \frac{(n_1 + n_2)!}{n_1! n_2!} \left(\frac{\lambda_1}{\lambda_1 + \lambda_2}\right)^{n_1} \left(\frac{\lambda_2}{\lambda_1 + \lambda_2}\right)^{n_2} \\ & + \frac{1}{\lambda_2 + \lambda_3} \sum_{n_2=0}^{k_2-1} \sum_{n_3=0}^{k_3-1} \frac{(n_2 + n_3)!}{n_2! n_3!} \left(\frac{\lambda_2}{\lambda_2 + \lambda_3}\right)^{n_2} \left(\frac{\lambda_3}{\lambda_2 + \lambda_3}\right)^{n_3} \\ & + \frac{1}{\lambda_3 + \lambda_1} \sum_{n_3=0}^{k_3-1} \sum_{n_1=0}^{k_1-1} \frac{(n_3 + n_1)!}{n_3! n_1!} \left(\frac{\lambda_3}{\lambda_3 + \lambda_1}\right)^{n_3} \left(\frac{\lambda_1}{\lambda_3 + \lambda_1}\right)^{n_1} \\ & - 2 \frac{1}{\lambda_1 + \lambda_2 + \lambda_3} \sum_{n_1=0}^{k_1-1} \sum_{n_2=0}^{k_2-1} \sum_{n_3=0}^{k_3-1} \frac{(n_1 + n_2 + n_3)!}{n_1! n_2! n_3!} a_1^{n_1} a_2^{n_2} a_3^{n_3}. \end{aligned} \quad (3.2.5)$$

□



**Figure C.2:** The decay rate achieved as a function of the number of packets sent via path 1 in the canonical two-path scenario. Both the paths are assumed to have gamma delays with the same mean but different variance (achieved by scaling up the parameters of the gamma distribution). The gamma distribution has parameters 40, 2 and the scale-up parameters are shown in the legend. The data size in this case is 30. The arrival rate is assumed to be 0.15. The decay rate is maximised at the corresponding proportional allocation. We also observe that smaller variance gives higher decay rate.

### C.3 RIGID ALLOCATION

Here we present another example of rigid allocation when the path latencies are not exponentially distributed. In Figure C.2, we plot the decay rate as a function of the number of packets sent via path 1 in the canonical two-path scenario. We observe that smaller variance yields higher decay rate.

## D.1 EXAMPLE OF SCHEDULING

**Example D.1.1.** Consider a 2-server queueing system with buffer lengths  $K_1, K_2$  and probability vector  $\pi = (\pi_1, 1 - \pi_1)$  for scheduling. Let  $X = (X_1, X_2)$  denote the queue lengths as before. Writing  $p_t(x, y) := P(X(t) = x, L(t) = y)$  for  $x = (x_1, x_2)$ , the time evolution of  $p_t$  is captured through the following CMEs

$$\frac{d}{dt}p_t(x, y) = \begin{cases} \gamma(x)p_t(x, y-1) + \alpha_1(x_1-1, x_2)p_t(x_1-1, x_2, y) \\ + \alpha_2(x_1, x_2-1)p_t(x_1, x_2-1, y) \\ + \beta_1(x_1+1, x_2)p_t(x_1+1, x_2, y) \\ + \beta_2(x_1, x_2+1)p_t(x_1, x_2+1, y) \\ - (\alpha_1(x) + \alpha_2(x) + \beta_1(x) + \beta_2(x) + \gamma(x)) p_t(x, y) & \text{if } x_1, x_2, y \geq 1, \\ \\ \alpha_1(x_1-1, x_2)p_t(x_1-1, x_2, y) \\ + \alpha_2(x_1, x_2-1)p_t(x_1, x_2-1, y) \\ + \beta_1(x_1+1, x_2)p_t(x_1+1, x_2, y) \\ + \beta_2(x_1, x_2+1)p_t(x_1, x_2+1, y) \\ - (\alpha_1(x) + \alpha_2(x) + \beta_1(x) + \beta_2(x) + \gamma(x)) p_t(x, y) & \text{if } x_1, x_2 \geq 1, y = 0, \\ \\ \alpha_2(x_1, x_2-1)p_t(x_1, x_2-1, y) \\ + \beta_1(x_1+1, x_2)p_t(x_1+1, x_2, y) \\ + \beta_2(x_1, x_2+1)p_t(x_1, x_2+1, y) \\ - (\alpha_1(x) + \alpha_2(x) + \beta_1(x) + \beta_2(x)) p_t(x, y) & \text{if } x_1 = y = 0, x_2 \geq 1, \\ \\ \alpha_1(x_1-1, x_2)p_t(x_1-1, x_2, y) \\ + \beta_1(x_1+1, x_2)p_t(x_1+1, x_2, y) \\ + \beta_2(x_1, x_2+1)p_t(x_1, x_2+1, y) \\ - (\alpha_1(x) + \alpha_2(x) + \beta_1(x) + \beta_2(x)) p_t(x, y) & \text{if } x_1 \geq 1, x_2 = y = 0, \\ \\ \beta_1(x_1+1, x_2)p_t(x_1+1, x_2, y) \\ + \beta_2(x_1, x_2+1)p_t(x_1, x_2+1, y) \\ - (\alpha_1(x) + \alpha_2(x) + \beta_1(x) + \beta_2(x)) p_t(x, y) & \text{if } x_1 = x_2 = 0, y \geq 0. \end{cases}$$

The above system of equations is infinite-dimensional, because the total loss process is unbounded. Therefore, as done before, we marginalise out  $L$  from the above, and solve

the reduced system of  $(K_1 + 1) \times (K_2 + 1)$  equations. Write  $q_t(x) := P(X(t) = x)$  for  $x = (x_1, x_2)$ . We need to solve only the following set of ODEs

$$\frac{d}{dt}q_t(x) = \begin{cases} \mu_1 q_t(1, 0) + \mu_2 q_t(0, 1) - \lambda q_t(0, 0) & \text{if } x = (0, 0), \\ \lambda \pi_2 q_t(0, x_2 - 1) + \mu_1 q_t(1, x_2) + \mu_2 q_t(0, x_2 + 1) \\ - (\lambda + \mu_2) q_t(0, x_2) & \text{if } x_1 = 0, x_2 \in [K_2 - 1], \\ \lambda \pi_1 q_t(x_1 - 1, 0) + \mu_1 q_t(x_1 + 1, 0) + \mu_2 q_t(x_1, 1) \\ - (\lambda + \mu_1) q_t(x_1, 0) & \text{if } x_2 = 0, x_1 \in [K_1 - 1], \\ \lambda \pi_1 q_t(x_1 - 1, x_2) + \lambda \pi_2 q_t(x_1, x_2 - 1) \\ + \mu_1 q_t(x_1 + 1, x_2) + \mu_2 q_t(x_1, x_2 + 1) \\ - (\lambda + \mu_1 + \mu_2) q_t(x_1, x_2) & \text{if } x_1 \in [K_1 - 1], x_2 \in [K_2 - 1], \\ \lambda q_t(x_1 - 1, K_2) + \lambda \pi_2 q_t(x_1, K_2 - 1) \\ + \mu_1 q_t(x_1 + 1, K_2) - (\lambda + \mu_1 + \mu_2) q_t(x_1, K_2) & \text{if } x_2 = K_2, x_1 \in [K_1 - 1], \\ \lambda \pi_2 q_t(0, K_2 - 1) + \mu_1 q_t(1, K_2) \\ - (\lambda + \mu_2) q_t(0, K_2) & \text{if } x_2 = K_2, x_1 = 0, \\ \lambda \pi_1 q_t(K_1 - 1, x_2) + \lambda q_t(K_1, x_2 - 1) \\ + \mu_2 q_t(K_1, x_2 + 1) - (\lambda + \mu_1 + \mu_2) q_t(K_1, x_2) & \text{if } x_1 = K_1, x_2 \in [K_2 - 1], \\ \lambda \pi_1 q_t(K_1 - 1, 0) + \mu_2 q_t(K_1, 1) \\ - (\lambda + \mu_1) q_t(K_1, 0) & \text{if } x_1 = K_1, x_2 = 0, \\ \lambda \pi_1 q_t(x_1 - 1, x_2) + \lambda \pi_2 q_t(x_1, x_2 - 1) \\ - (\mu_1 + \mu_2) q_t(K_1, K_2) & \text{if } x_1 = K_1, x_2 = K_2. \end{cases}$$

We solve the above  $(K_1 + 1) \times (K_2 + 1)$  ODEs and then numerically find  $\pi_{\text{opt}}$  as a function of the solution (recall (6.2.1)).

## D.2 SCALING LIMIT

### D.2.1 Properties of the limit

Before we prove Lemma 6.4.2, we prove the following uniqueness and smoothness properties of the solution of the integral equation (6.4.3).



*Proof of Lemma 6.4.1.* Following Martin and Suhov (1999) and Mukhopadhyay, Karthik, and Mazumdar (2016), define the operators

$$\begin{aligned}\mathbb{H}_{0,i}(u) &:= 0, \quad \forall i \in [M], \\ \mathbb{H}_{n,i}(u) &:= \frac{\lambda}{\nu_i} \left( \zeta(u_{n-1,i})^{S_i} - \zeta(u_{n,i})^{S_i} \right) \prod_{j \in [M] \setminus \{i\}} \zeta(u_{\theta_j(i,n-1),j})^{S_j} - \frac{\mu_i}{m} (\zeta(u_{n,i}) - \zeta(u_{n+1,i})),\end{aligned}\tag{4.2.1}$$

where  $\zeta(x) = \max(0, \min(x, 1))$ . Consider the solutions of the integral equation

$$w(t) = w(0) + \int_0^t \mathbb{H}(w(s)) ds.\tag{4.2.2}$$

Note that the  $\mathbb{H}(u)$  is defined for  $u \in (\mathbb{R}^K)^M$ . The operators  $\mathbb{H}(u)$  and  $\mathbb{F}(u)$  agree if  $u \in \mathcal{Z}^M$ . Therefore, the two systems (6.4.3) and (4.2.2) yield identical solutions in  $\mathcal{Z}^M$ . Moreover, if  $w(0) = u \in \mathcal{Z}^M$ , then the solution of the modified system (4.2.2) remains within  $\mathcal{Z}^M$  (see Mukhopadhyay, Karthik, and Mazumdar (2016) for similar arguments). In order to show uniqueness of solutions to (6.4.3) in  $\mathcal{Z}^M$ , it suffices to show that solutions to (4.2.2) are unique in  $(\mathbb{R}^K)^M$ . Therefore, we extend the norm  $\rho$  defined in (6.4.1) to  $(\mathbb{R}^K)^M$ .

Following the same line of argument as in Mukhopadhyay, Karthik, and Mazumdar (2016), we can find constants  $a, b \in \mathbb{R}_+$  such that

$$\begin{aligned}\rho_{(\mathbb{R}^K)^M}(\mathbb{H}(u), \mathbb{H}(u)) &\leq a, \\ \rho_{(\mathbb{R}^K)^M}(\mathbb{H}(u), \mathbb{H}(v)) &\leq b \rho_{(\mathbb{R}^K)^M}(u, v).\end{aligned}$$

The uniqueness of the solution follows by virtue of the above, and using Picard's iterative approximation method, because the space  $(\mathbb{R}^K)^M$  is complete under the metric defined in (6.4.1) (extended to  $(\mathbb{R}^K)^M$ ). □

**Lemma D.2.1.** *The partial derivatives*

$$\frac{\partial}{\partial u_{n,i}} z(t, u), \quad \frac{\partial^2}{\partial u_{n,i}^2} z(t, u), \quad \text{and} \quad \frac{\partial^2}{\partial u_{n,j} \partial u_{n,i}} z(t, u)$$

exist for all  $u \in \mathcal{Z}^M$ , and are uniformly bounded above as follows

$$\left| \frac{\partial}{\partial u_{n,i}} z(t, u) \right| \leq \exp(at),\tag{4.2.3}$$

$$\left| \frac{\partial^2}{\partial u_{n,i}^2} z(t, u) \right|, \left| \frac{\partial^2}{\partial u_{n,j} \partial u_{n,i}} z(t, u) \right| \leq \exp(bt),\tag{4.2.4}$$

for some constants  $a, b \in \mathbb{R}_+$ .

*Proof.* The proof follows along the same line of argument as in Mukhopadhyay, Karthik, and Mazumdar (2016, Lemma B.2) and Martin and Suhov (1999, Lemma 3.2) if we set

$$a := \frac{2\lambda \sum_{i=1 \in [M]} S_i}{\min_{i \in [M]} \nu_i} + \frac{2 \max_{i \in [M]} \mu_i}{m},\tag{4.2.5}$$

$$b := \sum_{i \in [M]} S_i + 2a.\tag{4.2.6}$$

Please note the that above bounds can be made tighter, but for our purposes, they suffice.  $\square$

*Proof of Lemma 6.4.2.* Let  $f \in \mathcal{C}_D$ . Then, for each  $i \in [M]$ , we have

$$\begin{aligned} \lim_{N_i \rightarrow \infty} N_i \left( f(u + \frac{1}{N_i} e_{n,i}) - f(u) \right) &= \frac{\partial}{\partial u_{n,i}} f(u), \\ \lim_{N_i \rightarrow \infty} N_i \left( f(u - \frac{1}{N_i} e_{n,i}) - f(u) \right) &= \frac{\partial}{\partial u_{n,i}} f(u), \end{aligned}$$

uniformly in  $u \in \mathcal{Z}^M$ . Therefore, from (6.4.2), we get

$$\begin{aligned} A_N f(u) &\rightarrow \sum_{i=1}^M \sum_{n=1}^K \frac{\lambda}{v_i} \left( (u_{n-1,i})^{S_i} - (u_{n,i})^{S_i} \right) \prod_{j \in [M] \setminus \{i\}} (u_{\theta_j(i,n-1),j})^{S_j} \frac{\partial}{\partial u_{n,i}} f(u) \\ &\quad - \sum_{i=1}^M \sum_{n=1}^K \frac{\mu_i}{m} (u_{n,i} - u_{n+1,i}) \frac{\partial}{\partial u_{n,i}} f(u), \end{aligned} \quad (4.2.7)$$

as  $N \rightarrow \infty$  in the light of F1 and F2. The right hand side of the above equation can be rearranged as follows

$$\sum_{i=1}^M \sum_{n=1}^K \left[ \frac{\lambda}{v_i} \left( (u_{n-1,i})^{S_i} - (u_{n,i})^{S_i} \right) \prod_{j \in [M] \setminus \{i\}} (u_{\theta_j(i,n-1),j})^{S_j} - \frac{\mu_i}{m} (u_{n,i} - u_{n+1,i}) \right] \frac{\partial}{\partial u_{n,i}} f(u),$$

which is identical with

$$\left. \frac{d}{dt} f(z(t, u)) \right|_{t=0},$$

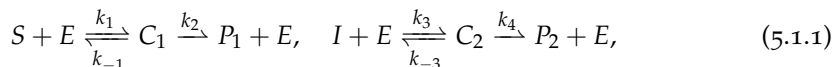
where  $z$  is the solution of the integral equation (6.4.3) with  $z(0) = u$ .  $\square$

## SUPPLEMENTARY MATERIAL TO CHAPTER 7

## E.1 ENZYME-SUBSTRATE-INHIBITOR SYSTEM

Inhibitors are compounds that diminish the rate of enzyme-catalysed reactions. They form complexes with the enzymes that exhibit a wide variety of catalytic properties. The most common type of inhibition is the competitive inhibition, where the inhibitor *competes* with the substrate in that it binds to the same site on the enzyme as the substrate (Cornish-Bowden 2004, Chapter 4). Interestingly, similar inhibitory behaviour is also observed when more than one substrates that bind with the same enzyme are present. This is commonly observed in many industrial applications. Each substrate competes with other substrates for the same catalytic site and inhibit each others' enzymatic reactions. There are also other variants of ESI system depending on the nature of competitiveness, such as the uncompetitive inhibition system, mixed inhibition system, substrate inhibition system etc. (Cornish-Bowden 2004, Chapter 4).

A fully competitive ESI system is described by the following set of chemical reactions (L. A. Segel 1988)



where  $C_1, C_2$  are respectively the substrate-enzyme and the inhibitor-enzyme complexes;  $P_1$ , and  $P_2$  are the respective products; and  $S, E$  are the substrate and free enzyme, as before. Classically, the ESI system has been studied using ODEs for the concentrations of the various species, as was done for the MM system. In order to specify our stochastic model, let  $X_S, X_I, X_E, X_{C_1}, X_{C_2}, X_{P_1}$ , and  $X_{P_2}$  denote the copy numbers of molecules of the substrates  $S$ , the inhibitors  $I$ , the enzymes  $E$ , the enzyme-substrate complex  $C_1$ , the inhibitor-enzyme complex  $C_2$ , and the products  $P_1, P_2$  respectively. As done in Section 7.3, we assume a Markovian dynamics for these copy numbers along with law of mass-action. We also introduce the scaled processes and the necessary exponents

$$\begin{aligned} X_i(t) &= N^{\alpha_i} Z_i^N(t), \text{ for } i = S, I, E, C_1, C_2, P_1, P_2, \\ \text{and } \kappa'_k &= N^{\beta_k} \kappa_k, \text{ for } k = 1, -1, 3, -3, 2, 4, \end{aligned} \quad (5.1.2)$$

**STANDARD QSSA (SQSSA) FOR THE ESI SYSTEM** The sQSSA for the ESI system is analogous to that for the MM enzyme kinetics described in Section 7.4. Here, one assumes both the enzyme-substrate complex  $C_1$  and the inhibitor-enzyme complex  $C_2$  reach a steady-state quickly after a brief transient phase while the other species still remain transient. Therefore, one sets  $\frac{d}{dt}[C_1] \approx 0$  and  $\frac{d}{dt}[C_2] \approx 0$ . Consider the following scaling exponents

$$\begin{aligned} \alpha_S &= \alpha_I = \alpha_{P_1} = \alpha_{P_2} = 1, & \alpha_E &= \alpha_{C_1} = \alpha_{C_2} = 0, \\ \beta_1 &= \beta_3 = 0, & \beta_{-1} &= \beta_{-3} = \beta_2 = \beta_4 = 1, \end{aligned} \quad (5.1.3)$$

to obtain the stochastic sQSSA, which is analogous to its deterministic counterpart (Pedersena, Bersanib, and Bersanic 2006; L. A. Segel 1988)

$$\dot{Z}_S(t) = -\frac{\kappa_2 M Z_S(t)}{\kappa_M^{(1)} \left(1 + \frac{Z_I(t)}{\kappa_M^{(2)}}\right) + Z_S(t)}, \quad (5.1.4)$$

$$\dot{Z}_I(t) = -\frac{\kappa_4 M Z_I(t)}{\kappa_M^{(2)} \left(1 + \frac{Z_S(t)}{\kappa_M^{(1)}}\right) + Z_I(t)}, \quad (5.1.5)$$

where  $M := Z_E^N(t) + Z_{C_1}^N(t) + Z_{C_2}^N(t)$ ,  $\kappa_M^{(1)} = (\kappa_{-1} + \kappa_2)/\kappa_1$  and  $\kappa_M^{(2)} = (\kappa_{-3} + \kappa_4)/\kappa_3$ . Detailed calculations are not shown for reasons of brevity. With regards to the validity of the sQSSA, the following conditions were proposed by L. A. Segel (1988)

$$[E_0] \ll \kappa_M^{(2)} \left(1 + \frac{[S_0]}{\kappa_M^{(1)}}\right) + [I_0] \quad \text{and} \quad [E_0] \ll \kappa_M^{(1)} \left(1 + \frac{[I_0]}{\kappa_M^{(2)}}\right) + [S_0], \quad (5.1.6)$$

where  $[E_0] := [E] + [C_1] + [C_2]$ ,  $[S_0] := [S] + [C_1] + [P_1]$  and  $[I_0] := [I] + [C_2] + [P_2]$  describe the conservation laws in the system. As done in Section 7.4, rewriting (5.1.6) in terms of species copy numbers, we see that the left hand sides of both inequalities in (5.1.6) correspond to  $Z_E + Z_{C_1} + Z_{C_2}$ , which is order 1. On the other hand,  $\kappa_M^{(2)} \left(1 + \frac{[S_0]}{\kappa_M^{(1)}}\right) + [I_0]$  simplifies to

$$N \left( \frac{\kappa_{-3} + \kappa_4}{\kappa_3} + \frac{(\kappa_{-3} + \kappa_4)\kappa_1}{\kappa_3(\kappa_{-1} + \kappa_2)} (Z_S + Z_{P_1}) + (Z_I + Z_{P_2}) \right) + \frac{(\kappa_{-3} + \kappa_4)\kappa_1}{\kappa_3(\kappa_{-1} + \kappa_2)} Z_{C_1} + Z_{C_2},$$

which is of order  $N$ . In a similar fashion, the quantity  $\kappa_M^{(1)} \left(1 + \frac{[I_0]}{\kappa_M^{(2)}}\right) + [S_0]$  can be simplified to  $N \left( \frac{\kappa_{-1} + \kappa_2}{\kappa_1} + \frac{(\kappa_{-1} + \kappa_2)\kappa_3}{\kappa_1(\kappa_{-3} + \kappa_4)} (Z_I + Z_{P_2}) + (Z_S + Z_{P_1}) \right) + \frac{(\kappa_{-1} + \kappa_2)\kappa_3}{\kappa_1(\kappa_{-3} + \kappa_4)} Z_{C_2} + Z_{C_1}$ , which is also of order  $N$ . Therefore, the condition (5.1.6) is included in the validity region for the stochastic sQSSA.

**TOTAL QSSA FOR THE ESI SYSTEM** In Pedersena, Bersanib, and Bersanic (2006), the authors propose the tQSSA for the ESI system. Following Borghans, De Boer, and L. A. Segel (1996), they define two new total substrates as follows

$$T_1 := S + C_1, \quad \text{and} \quad T_2 := I + C_2. \quad (5.1.7)$$

Applying the usual quasi-steady state approximation  $\frac{d}{dt}[C_1] \approx 0$  and  $\frac{d}{dt}[C_2] \approx 0$ , and assuming  $[C_1] < [T_1]$ ,  $[C_2] < [T_2]$ , one can rewrite the system of ODEs in terms of  $T_1$  and  $T_2$ . This yields the tQSSA for the ESI system (Pedersena, Bersanib, and Bersanic 2006). The idea behind the tQSSA is to achieve high accuracy over a wider range of initial conditions in the sense that the proposed sufficient conditions for the validity of tQSSA are roughly satisfied in almost all practical situations. For tQSSA, we apply the following scalings

$$\begin{aligned} \alpha_S = \alpha_I = \alpha_{P_1} = \alpha_{P_2} = \alpha_E = \alpha_{C_1} = \alpha_{C_2} &= 1, \\ \beta_1 = \beta_3 = \beta_2 = \beta_4 &= 0, \quad \beta_{-1} = \beta_{-3} = 1. \end{aligned} \quad (5.1.8)$$

In order to specify the conservation laws, let  $m^N := Z_E^N(t) + Z_{C_1}^N(t) + Z_{C_2}^N(t)$ , and  $l_1^N := Z_{T_1}^N(t) + Z_{P_1}^N(t)$ ,  $l_2^N := Z_{T_2}^N(t) + Z_{P_2}^N(t)$ . We assume  $m^N \rightarrow m$ ,  $l_1^N \rightarrow l_1$ ,  $l_2^N \rightarrow l_2$  as  $N \rightarrow \infty$ . Now, define the cumulative processes

$$\mathbb{Z}_{C_1}^N(t) := \int_0^t Z_{C_1}^N(s) ds \quad \text{and} \quad \mathbb{Z}_{C_2}^N(t) := \int_0^t Z_{C_2}^N(s) ds,$$

so that  $\mathbb{Z}_{C_1}^N(t) + \mathbb{Z}_{C_2}^N(t) = m^N t - \int_0^t Z_E^N(s) ds$ . Then, given the constants  $\kappa_D^{(1)} := \kappa_{-1}/\kappa_1$ , and  $\kappa_D^{(2)} := \kappa_{-3}/\kappa_3$ , the stochastic tQSSA is given by

$$\begin{aligned} \dot{\mathbb{Z}}_{C_1} = & m - \left( m - \dot{\mathbb{Z}}_{C_1}(t) \left( 1 + \frac{\kappa_D^{(1)}}{Z_{T_1}(t) - \dot{\mathbb{Z}}_{C_1}(t)} \right) \right) \\ & \times \left( 1 + \frac{\kappa_D^{(2)}}{Z_{T_2}(t) - m - \dot{\mathbb{Z}}_{C_1}(t) \left( 1 + \frac{\kappa_D^{(1)}}{Z_{T_1}(t) - \dot{\mathbb{Z}}_{C_1}(t)} \right)} \right), \end{aligned}$$

which implies the steady-state concentration of the substrate-enzyme complex is found by solving the following cubic equation for a positive root

$$\begin{aligned} p_3^{(1)}(\dot{\mathbb{Z}}_{C_1}) := & -(\kappa_D^{(1)} - \kappa_D^{(2)})(\dot{\mathbb{Z}}_{C_1})^3 \\ & + \left( (m + \kappa_D^{(1)} + Z_{T_1}(t))(\kappa_D^{(1)} - \kappa_D^{(2)}) - (Z_{T_1}(t)\kappa_D^{(2)} + Z_{T_2}(t)\kappa_D^{(1)}) \right) (\dot{\mathbb{Z}}_{C_1})^2 \\ & + \left( -m(\kappa_D^{(1)} - \kappa_D^{(2)}) + \kappa_D^{(2)}(m + \kappa_D^{(1)}) + (Z_{T_1}(t)\kappa_D^{(2)} + Z_{T_2}(t)\kappa_D^{(1)}) \right) Z_{T_1}(t)\dot{\mathbb{Z}}_{C_1} \\ & - m\kappa_D^{(2)}(Z_{T_1}(t))^2. \end{aligned} \quad (5.1.9)$$

An analogous third degree polynomial can be written for  $\dot{\mathbb{Z}}_{C_2}$ . Finally, the tQSSA for the totals is expressed as follows:

$$\dot{\mathbb{Z}}_{T_1}(t) = -\kappa_2 \dot{\mathbb{Z}}_{C_1}, \quad \text{and} \quad \dot{\mathbb{Z}}_{T_2}(t) = -\kappa_4 \dot{\mathbb{Z}}_{C_2}, \quad (5.1.10)$$

where  $\dot{\mathbb{Z}}_{C_1}$ , and  $\dot{\mathbb{Z}}_{C_2}$  satisfy their respective cubic equations (Pedersen, Bersani, and Bersani 2006).

Interestingly, when the substrate and the inhibitor have identical affinity towards the enzyme in that  $\kappa_D^{(1)} = \kappa_D^{(2)} = \kappa_D$ , the third degree polynomial  $p_3^{(1)}$  reduces to a second degree polynomial, allowing for simpler computations. In that case, the tQSSA limiting ODEs are given by

$$\begin{aligned} \dot{\mathbb{Z}}_{T_1}(t) = & -\kappa_2 \frac{Z_{T_1}(t)(Z_T(t) + \kappa_D + m)}{2Z_T(t)} \left( 1 - \sqrt{1 - \frac{4mZ_T(t)}{(Z_T(t) + \kappa_D + m)^2}} \right), \\ \dot{\mathbb{Z}}_{T_2}(t) = & -\kappa_4 \frac{Z_{T_2}(t)(Z_T(t) + \kappa_D + m)}{2Z_T(t)} \left( 1 - \sqrt{1 - \frac{4mZ_T(t)}{(Z_T(t) + \kappa_D + m)^2}} \right), \end{aligned} \quad (5.1.11)$$

where  $Z_T(t) := Z_{T_1}(t) + Z_{T_2}(t)$  is the total of the substrate, the inhibitor, the substrate-enzyme complex and the inhibitor-enzyme complex. Note that the limiting equations

given in (5.1.10) and (5.1.11) are analogous to their deterministic counterparts with the exception that we have  $\kappa_D^{(1)}, \kappa_D^{(2)}$  instead of the MM type constants  $\kappa_M^{(1)}, \kappa_M^{(2)}$ . The reason behind this discrepancy is that the propensities of the product formations are of order  $N$ , which are slower than the other reactions leading to the disappearance of the constants  $\kappa_2$  and  $\kappa_4$ .

Regarding the validity of the tQSSA, the following sufficient condition was proposed by Pedersen, Bersanib, and Bersanic (2006)

$$\max\left\{\frac{k_2 C_1([T_0^{(1)}], [T_0^{(2)}])}{[T_0^{(1)}]}, \frac{k_4 C_2([T_0^{(1)}], [T_0^{(2)}])}{[T_0^{(2)}]}\right\} \max\left\{\frac{C_1([T_0^{(1)}], [T_0^{(2)}])}{k_1 [E_0] [T_0^{(1)}]}, \frac{C_2([T_0^{(1)}], [T_0^{(2)}])}{k_3 [E_0] [T_0^{(2)}]}\right\} \ll 1, \quad (5.1.12)$$

where, as before,  $[E_0] := [E] + [C_1] + [C_2]$ ,  $[T_0^{(1)}] := [S] + [C_1]$  and  $[T_0^{(2)}] := [I] + [C_2]$ , and  $C_1([T_0^{(1)}], [T_0^{(2)}])$ , and  $C_2([T_0^{(1)}], [T_0^{(2)}])$  are the steady-state concentrations of the substrate-enzyme complex and the inhibitor-enzyme complex treated as functions of the initial conditions  $[T_0^{(1)}], [T_0^{(2)}]$ . Since the quantities  $C_1, C_2$  are to be obtained as a positive root to a cubic equation analogous to (5.1.9), a direct comparison with the stochastic validity conditions is cumbersome. However, for the special case of identical affinity, we can simplify the equations and do a qualitative comparison. When the substrate and the inhibitor exhibit identical affinity, the quantities  $C_1$  and  $C_2$  admit the following relatively simpler expressions (Pedersen, Bersanib, and Bersanic 2006)

$$\begin{aligned} C_1([T_0^{(1)}], [T_0^{(2)}]) &= \frac{[T_0^{(1)}] ([T_0] + K_D + [E_0])}{2[T_0]} \left(1 - \sqrt{1 - \frac{4[E_0][T_0]}{([T_0] + K_D + [E_0])^2}}\right), \\ C_2([T_0^{(1)}], [T_0^{(2)}]) &= \frac{[T_0^{(2)}] ([T_0] + K_D + [E_0])}{2[T_0]} \left(1 - \sqrt{1 - \frac{4[E_0][T_0]}{([T_0] + K_D + [E_0])^2}}\right), \end{aligned} \quad (5.1.13)$$

where  $K_D = k_{-1}/k_1$ ,  $[T_0] := [T_0^{(1)}] + [T_0^{(2)}]$ . Then, the sufficient condition proposed by Pedersen, Bersanib, and Bersanic (2006) in (5.1.12) can be rewritten as

$$[E_0] \gg \frac{k_2}{k_1} \frac{([T_0] + K_D + [E_0])^2}{4[T_0]^2} \left(1 - \sqrt{1 - \frac{4[E_0][T_0]}{([T_0] + K_D + [E_0])^2}}\right)^2,$$

which allows for a direct comparison with our stochastic system. As done in Section 7.5, if we convert the concentrations appearing above to molecular species copy numbers in our stochastic system, we can immediately see that the left hand side of the above inequality is of order  $N$ . On the other hand, the right hand side is of order 1. To see this, note that the quantity  $\frac{([T_0] + K_D + [E_0])^2}{4[T_0]^2}$  corresponds to  $\frac{(Z_T(t) + \kappa_D + m)^2}{(Z_T(t))^2}$  and therefore, is of order 1. Similarly, the quantity under the square root sign is also of order 1 for our choice of the scaling exponents. Therefore, the inequality above is satisfied in our stochastic set-up. The validity region for the deterministic tQSSA for the ESI set-up is therefore included in the validity region of the stochastic tQSSA.

## SUPPLEMENTARY MATERIAL TO CHAPTER 8

## F.1 HYPERGEOMETRIC MOMENTS

Here we compute various (conditional) moments that are useful for our derivations. The following moments are computed keeping Remark 8.3.1 in mind. In a straightforward fashion we get,

$$\begin{aligned}
\mathbb{E}[(X_{SI,i})_3 \mid \mathcal{F}_{t-}] &= \frac{(k)_3(X_{SI})_3}{(X_{S\bullet})_3}, \\
\mathbb{E}[(X_{SI,i})_2 \mid \mathcal{F}_{t-}] &= \frac{(k)_2(X_{SI})_2}{(X_{S\bullet})_2}, \\
\mathbb{E}[X_{SI,i}^3 \mid \mathcal{F}_{t-}] &= \frac{(k)_3(X_{SI})_3}{(X_{S\bullet})_3} + 3\frac{(k)_2(X_{SI})_2}{(X_{S\bullet})_2} + k\frac{X_{SI}}{X_{S\bullet}}, \\
\mathbb{E}[X_{SI,i}(X_{SS,i})_2 \mid \mathcal{F}_{t-}] &= \frac{(k)_3X_{SI}(X_{SS})_2}{(X_{S\bullet})_3}, \\
\mathbb{E}[(X_{SI,i})_2X_{SS,i} \mid \mathcal{F}_{t-}] &= \frac{(k)_3(X_{SI})_2X_{SS}}{(X_{S\bullet})_3}, \\
\mathbb{E}[X_{SI,i}X_{SS,i} \mid \mathcal{F}_{t-}] &= \frac{(k)_2X_{SI}X_{SS}}{(X_{S\bullet})_2}.
\end{aligned}$$

## F.2 CONVERGENCE OF THE QUADRATIC VARIATION PROCESS

*Proof of Lemma 8.3.1.* To show convergence of the matrix random process  $\langle M \rangle(t)$  to  $V(t)$ , we show element-wise convergence of respective components. The general strategy to prove convergence for these components remains the same. To save the reader from repetitive lines of argument, we only demonstrate here the strategy for establishing  $M_{SI}(t) \xrightarrow{P} V_{SI}(t)$ . Remaining assertions follow similarly.

**COMPUTATION OF  $\langle M_{SI} \rangle$**  The process  $M_{SI}$  jumps only if infection of a node occurs. Therefore, the predictable quadratic variation is computed as follows

$$\langle M_{SI} \rangle(t) = \langle n^{-1/2} M'_{SI} \rangle(t) = \int_0^t \sum_k \frac{1}{n} \sum_{i \in S_k} \beta X_{SI,i} (X_{SS,i} - X_{SI,i})^2 ds.$$

Now, for a randomly selected  $i \in S_k$ , we seek to find the (conditional) moments  $\mathbb{E}[X_{SI,i}(X_{SS,i} - X_{SI,i})^2 \mid \mathcal{F}_{t-}]$ . Define  $C_h^k : \mathcal{T} \rightarrow \mathbb{R}$  as  $C_h^k(t) := \mathbb{E}[X_{SI,i}(t)(X_{SS,i}(t) - X_{SI,i}(t))^2 \mid \mathcal{F}_{t-}]$ . Following the computations in Appendix F.1, we get

$$\begin{aligned} C_h^k(t) &= \frac{(k)_3 X_{SI}}{(X_S \bullet)_3} [(X_{SS})_2 - 2(X_{SI} - 1)X_{SS} + (X_{SI} - 1)_2] \\ &\quad - \frac{(k)_2 X_{SI}}{(X_S \bullet)_2} [X_{SS} - 3(X_{SI} - 1)] + k \frac{X_{SI}}{X_S \bullet}. \end{aligned}$$

To approximate the hypergeometric moments by corresponding multinomial moments, define the multinomial compensator  $C_m^k : \mathcal{T} \times [\xi, 2\partial\psi(1)] \rightarrow \mathbb{R}$  as

$$\begin{aligned} C_m^k(t, z) &:= \frac{(k)_3 n^{-3} X_{SI}}{z^3} (X_{SS}^2 - 2X_{SI}X_{SS} + X_{SI}^2) \\ &\quad - \frac{(k)_2 n^{-2} X_{SI}}{z^2} (X_{SS} - 3X_{SI}) + \frac{kn^{-1} X_{SI}}{z} \\ &= \frac{(k)_3 n^{-3} X_{SI} (X_{SS} - X_{SI})^2}{z^3} - \frac{(k)_2 n^{-2} X_{SI} (X_{SS} - 3X_{SI})}{z^2} + \frac{kn^{-1} X_{SI}}{z}. \end{aligned}$$

Please observe that there exists an  $L > 0$  such that

$$C_m^k(t, z(t)) \leq Lk^3, \quad (6.2.1)$$

uniformly in  $n$ . This holds because  $n^{-1}X_{SI}$  and  $n^{-1}X_{SS}$  are uniformly bounded above by virtue of Remark 8.3.2 and  $z$  is bounded away from zero, by definition. The function  $C_m^k(t, z(t))$  is also Lipschitz continuous in  $z$ . Now recall the definition of  $v$  from (8.3.5) and define

$$\begin{aligned} \Delta(t) &:= \sum_k \frac{1}{n} \sum_{i \in S_k} \beta X_{SI,i}(t) (X_{SS,i}(t) - X_{SI,i}(t))^2 - v_{SI}(x(t), \theta(t)) \\ &= \sum_k \frac{1}{n} \sum_{i \in S_k} \beta X_{SI,i}(t) (X_{SS,i}(t) - X_{SI,i}(t))^2 - v_{SI}(n^{-1}X(t), \theta(t)) \\ &\quad + v_{SI}(n^{-1}X(t), \theta(t)) - v_{SI}(x(t), \theta(t)) \\ &= \Delta_1(t) + \Delta_2(t), \end{aligned}$$

where  $\Delta_1(t) := \sum_k \frac{1}{n} \sum_{i \in S_k} \beta X_{SI,i}(t) (X_{SS,i}(t) - X_{SI,i}(t))^2 - v_{SI}(n^{-1}X(t), \theta(t))$ , and  $\Delta_2(t) := v_{SI}(n^{-1}X(t), \theta(t)) - v_{SI}(x(t), \theta(t))$ . In order to show  $\langle M_{SI} \rangle \xrightarrow{P} V_{SI}$ , it suffices to show  $\sup_{t \in \mathcal{T}_0} |\Delta(t)| \xrightarrow{P} 0$ . We achieve this by separately showing  $\sup_{t \in \mathcal{T}_0} |\Delta_1(t)| \xrightarrow{P} 0$  and  $\sup_{t \in \mathcal{T}_0} |\Delta_2(t)| \xrightarrow{P} 0$ .



CONVERGENCE OF  $\Delta_1(t)$  See that

$$\begin{aligned}
\Delta_1(t) &= \sum_k \frac{1}{n} \sum_{i \in S_k} \beta X_{SI,i} (X_{SS,i} - X_{SI,i})^2 - \beta \left[ \frac{n^{-3} X_{SI} (X_{SS} - X_{SI})^2}{\alpha_S^2} \frac{\partial^3 \psi(\theta)}{(\partial \psi(\theta))^3} \right. \\
&\quad \left. - \frac{n^{-2} X_{SI} (X_{SS} - 3X_{SI})}{\alpha_S} \frac{\partial^2 \psi(\theta)}{(\partial \psi(\theta))^2} + n^{-1} X_{SI} \right] \\
&= \sum_k \left[ \frac{1}{n} \sum_{i \in S_k} \beta X_{SI,i} (X_{SS,i} - X_{SI,i})^2 - \beta \left\{ \frac{n^{-3} X_{SI} (X_{SS} - X_{SI})^2}{\alpha_S^2} \frac{(k)_3 \theta^k p_k}{(\theta \partial \psi(\theta))^3} \right. \right. \\
&\quad \left. \left. - \frac{n^{-2} X_{SI} (X_{SS} - 3X_{SI})}{\alpha_S} \frac{(k)_2 \theta^k p_k}{(\theta \partial \psi(\theta))^2} + n^{-1} X_{SI} \frac{k \theta p_k}{\theta \partial \psi(\theta)} \right\} \right] \\
&= \sum_k \left[ \frac{1}{n} \sum_{i \in S_k} \beta X_{SI,i} (X_{SS,i} - X_{SI,i})^2 - \beta \alpha_S p_k \theta^k C_m^k(t, \alpha_S \theta \partial \psi(\theta)) \right].
\end{aligned}$$

Define  $\Delta_1^{(k)}(t) := \frac{1}{n} \sum_{i \in S_k} \beta X_{SI,i} (X_{SS,i} - X_{SI,i})^2 - \beta \alpha_S p_k \theta^k C_m^k(t, \alpha_S \theta \partial \psi(\theta))$ . Our task boils down to showing that  $\sup_{t \in \mathcal{T}_0} |\sum_k \Delta_1^{(k)}(t)| \xrightarrow{P} 0$  as  $n \rightarrow \infty$ . We achieve this task in two steps. First we show that the tails of  $\sum_k \Delta_1^{(k)}(t)$  are negligible. Second, we show that each term  $\Delta_1^{(k)}(t)$  converges to zero uniformly in probability for a fixed  $k \in \mathbb{N}$ .

(STEP I) TAILS ARE NEGLIGIBLE Let us begin by showing that as  $N \rightarrow \infty$ ,

$$\sup_{n \in \mathbb{N}} \sup_{t \in \mathcal{T}_0} \left| \sum_{k > N} \Delta_1^{(k)}(t) \right| \xrightarrow{P} 0.$$

Observe that

$$\left| \frac{1}{n} \sum_{k > N} \sum_{i \in S_k} \beta X_{SI,i} (X_{SS,i} - X_{SI,i})^2 \right| \leq \frac{\beta}{n} \sum_{k > N} k^3 X_{S_k} \leq 2\beta \sum_{k > N} k^3 p_k, \quad (6.2.2)$$

because  $n^{-1} X_{S_k} \leq 2p_k$  for sufficiently large  $n$  in the light of Remark 8.3.2. Following Remark 8.3.2 and the bound on  $C_m^k$  from (6.2.1), we get

$$\left| \sum_{k > N} \beta \alpha_S p_k \theta^k C_m^k(t, \alpha_S \theta \partial \psi(\theta)) \right| \leq \beta L \sum_{k > N} k^3 p_k. \quad (6.2.3)$$

Therefore, we get  $\sup_{n \in \mathbb{N}} \sup_{t \in \mathcal{T}_0} |\sum_{k > N} \Delta_1^{(k)}(t)| \xrightarrow{P} 0$ , combining inequalities (6.2.2) and (6.2.3) in view of A3.

(STEP II) UNIFORM CONVERGENCE IN PROBABILITY FOR A FIXED  $k$  In addition to Step I, it is sufficient to show  $\sup_{t \in \mathcal{T}_0} |\Delta_1^{(k)}(t)| \xrightarrow{P} 0$  for an arbitrarily fixed  $k \in \mathbb{N}$  to justify  $\sup_{t \in \mathcal{T}_0} |\Delta_1(t)| \xrightarrow{P} 0$ . Observe that

$$|\Delta_1^{(k)}(t)| = \left| \frac{1}{n} \sum_{i \in S_k} \beta X_{SI,i} (X_{SS,i} - X_{SI,i})^2 - \beta \alpha_S p_k \theta^k C_m^k(t, \alpha_S \theta \partial \psi(\theta)) \right|$$

$$\leq \beta n^{-1} \left| \sum_{i \in S_k} X_{SI,i} (X_{SS,i} - X_{SI,i})^2 - X_{S_k} C_h^k(t) \right| \quad (6.2.4)$$

$$+ \beta n^{-1} X_{S_k} |C_h^k(t) - C_m^k(t, n^{-1} X_{S_\bullet})| \quad (6.2.5)$$

$$+ \beta |n^{-1} X_{S_k} C_m^k(t, n^{-1} X_{S_\bullet}) - \alpha_S p_k \theta^k C_m^k(t, n^{-1} X_{S_\bullet})| \quad (6.2.6)$$

$$+ \beta \alpha_S p_k \theta^k |C_m^k(t, n^{-1} X_{S_\bullet}) - C_m^k(t, \alpha_S \theta \partial \psi(\theta))|. \quad (6.2.7)$$

We show that each of the above summands converges uniformly in probability to zero.

Define the process  $\Delta_{1,1}^{(k)}(t) := \sum_{i \in S_k} X_{SI,i} (X_{SS,i} - X_{SI,i})^2 - X_{S_k} C_h^k(t)$ . Observe that  $\Delta_{1,1}^{(k)}(t)$  is a zero-mean, piecewise-constant, càdlàg martingale with paths in  $D$ . The jumps of  $\Delta_{1,1}^{(k)}(t)$  take place when a node of degree- $k$  gets infected. The quadratic variation of  $\Delta_{1,1}^{(k)}(t)$  is therefore the sum of its squared jumps

$$[\Delta_{1,1}^{(k)}](t) = \sum_{s \leq t} (\delta \Delta_{1,1}^{(k)}(s))^2 \leq k^6 n,$$

because the number of jumps can not exceed  $n$ . Therefore by Doob's martingale inequality we get  $\sup_{t \in \mathcal{T}_0} |n^{-1} \Delta_{1,1}^{(k)}(t)| \xrightarrow{P} 0$ , since  $E[[\Delta_{1,1}^{(k)}](t)] = E[(\Delta_{1,1}^{(k)}(t))^2] = O(n)$ . That is, the quantity in (6.2.4) converges uniformly in probability to zero.

For the term in (6.2.5), take into account  $n^{-1} X_{S_k} \leq 1$  and see that

$$\sup_{t \in \mathcal{T}_0} |C_h^k(t) - C_m^k(t, n^{-1} X_{S_\bullet})| \leq \frac{c_1 k^3}{X_{S_\bullet}(T) - 2},$$

for some  $c_1 > 0$ , because  $X_{S_\bullet}$  is non-increasing on  $\mathcal{T} = [0, T]$ . Therefore, by A1, the quantity in (6.2.5) converges to zero uniformly in probability.

Now observe that

$$\sup_{t \in \mathcal{T}_0} |n^{-1} X_{S_k} C_m^k(t, n^{-1} X_{S_\bullet}) - \alpha_S p_k \theta^k C_m^k(t, n^{-1} X_{S_\bullet})|$$

$$\leq L k^3 \sup_{t \in \mathcal{T}_0} |n^{-1} X_{S_k} - \alpha_S p_k \theta^k| \xrightarrow{P} 0,$$

by virtue of the bound on  $C_m^k$  in (6.2.1) and Jacobsen et al. (2016, Lemma 1(a)). Therefore, the term in (6.2.6) also converges to zero uniformly in probability.

Finally by virtue of Lipschitz continuity of  $C_m^k(t, z)$  in  $z$ , we get

$$\sup_{t \in \mathcal{T}_0} |C_m^k(t, n^{-1} X_{S_\bullet}) - C_m^k(t, \alpha_S \theta \partial \psi(\theta))| \leq c_2 \sup_{t \in \mathcal{T}_0} |n^{-1} X_{S_\bullet} - \alpha_S \theta \partial \psi(\theta)|,$$

for some  $c_2 > 0$ . Because  $\sup_{t \in \mathcal{T}_0} |n^{-1}X_{S_\bullet} - \alpha_S \theta \partial \psi(\theta)| \xrightarrow{\mathbb{P}} 0$  as shown in Jacobsen et al. (2016), we conclude that the term in (6.2.7) converges to zero uniformly in probability.

Having shown the terms in (6.2.4), (6.2.5), (6.2.6) and (6.2.7) converge to zero uniformly in probability, we establish that  $\sup_{t \in \mathcal{T}_0} |\Delta_1^{(k)}(t)| \xrightarrow{\mathbb{P}} 0$  uniformly in probability for any fixed  $k \in \mathbb{N}$ . Finally, by virtue of Step I and Step II, we obtain  $\sup_{t \in \mathcal{T}_0} |\Delta_1(t)| \xrightarrow{\mathbb{P}} 0$ .

**CONVERGENCE OF  $\Delta_2(t)$**  Note that  $v_{SI}(n^{-1}X, \theta)$  is Lipschitz continuous on its domain that we can take as  $(0, 1] \times [\zeta, 2\partial\psi(1)]^2 \times [\zeta, 1]$ , by Remark 8.3.2. Therefore,

$$\sup_{t \in \mathcal{T}_0} |v_{SI}(n^{-1}X, \theta) - v_{SI}(x, \vartheta)| \leq c_3 \sup_{t \in \mathcal{T}_0} \|(n^{-1}X, \theta) - (x, \vartheta)\|,$$

for some Lipschitz constant  $c_3 > 0$ . Since  $(x, \vartheta)$  is the solution of (8.2.7), with initial condition  $x(0) = \alpha$  and  $\vartheta(0) = 1$ , we get by virtue of Theorem 8.2.1,  $\sup_{t \in \mathcal{T}_0} |\Delta_2(t)| \xrightarrow{\mathbb{P}} 0$ .

**FINAL CONCLUSION** Since  $\sup_{t \in \mathcal{T}_0} |\Delta_1(t)| \xrightarrow{\mathbb{P}} 0$  and  $\sup_{t \in \mathcal{T}_0} |\Delta_2(t)| \xrightarrow{\mathbb{P}} 0$ , we conclude  $\sup_{t \in \mathcal{T}_0} |\Delta(t)| \xrightarrow{\mathbb{P}} 0$ , which is a sufficient condition for

$$\langle M_{SI} \rangle(t) \xrightarrow{\mathbb{P}} V_{SI}(t) := \int_0^t v_{SI}(x(s), \vartheta(s)) \, ds.$$

□

### F.3 INTERPRETATION OF THE $\mathbb{D}$ OPERATOR

*Proof of Lemma 8.5.1.* The probability that a randomly chosen node  $i$  is susceptible and is of degree  $k$  is given by  $\mathbb{P}(i \in S_k(t)) = n^{-1}X_S(0)\theta^k(t)p_k$ . The following is then immediate.

$$\mu_S(\theta(t)) = \sum_k k \mathbb{P}(i \in S_k(t) \mid i \in S(t)) = \frac{\sum_k k \theta^k(t) p_k}{\sum_k \theta^k(t) p_k} = \frac{\theta(t) \partial \psi(\theta(t))}{\psi(\theta(t))}.$$

In order to explicitly calculate  $\mu_S^{(r)}$ , it will be helpful to keep the dynamic construction of the graph in mind. In particular, we make use the neighbourhood distribution of a susceptible node given in Remark 8.3.1. Therefore,

$$\begin{aligned} \mu_S^{(r)}(\theta(t)) &= \frac{\sum_k (k-r) \mathbb{P}(i \in S_k(t)) \mathbb{E}[(X_{SI,i})_r \mid \mathcal{F}_{t-}]}{\sum_k \mathbb{P}(i \in S_k(t)) \mathbb{E}[(X_{SI,i})_r \mid \mathcal{F}_{t-}]} \\ &= \frac{\sum_k (k-r+1) \theta^k(t) p_k}{\sum_k (k-r) \theta^k(t) p_k} \\ &= \frac{\theta(t) \partial^{r+1} \psi(\theta(t))}{\partial^r \psi(\theta(t))}. \end{aligned}$$

The recurrence relation then follows in a straightforward manner.

$$\mathbb{D}^{r+1} \psi(\theta) = \frac{\theta \partial^{r+1} \psi(\theta)}{\partial^r \psi(\theta)} \times \frac{\psi(\theta)}{\theta \partial \psi(\theta)} \times \frac{\psi^{r-1}(\theta) \partial^r \psi(\theta)}{(\partial \psi(\theta))^r} = \frac{\mu_S^{(r)}(\theta)}{\mu_S(\theta)} \mathbb{D}^r \psi(\theta).$$

The convergence  $\mathbb{D}^r \psi(\theta) \xrightarrow{\mathbb{P}} \mathbb{D}^r \psi(\vartheta)$ , uniformly on  $\mathcal{T}$ , follows virtue of Theorem 8.2.1. This completes the proof.  $\square$

For our purposes, we only need  $\frac{\mu_S^{(1)}(\theta)}{\mu_S(\theta)} \xrightarrow{\mathbb{P}} \kappa(\vartheta) = \mathbb{D}^2 \psi(\vartheta)$ ,  $\frac{\mu_S^{(2)}(\theta)}{\mu_S(\theta)} \kappa(\theta) \xrightarrow{\mathbb{P}} \mathbb{D}^3 \psi(\vartheta)$ , and hence the interpretation in Section 8.2 as a limiting ratio follows. The two operators  $\mathbb{D}^2 \psi(\vartheta)$ , and  $\mathbb{D}^3 \psi(\vartheta)$  essentially allow us to correctly estimate various pair and triple counts in the large graph limit.

## SUPPLEMENTARY MATERIAL TO CHAPTER 9

## G.1 DERIVATIONS FOR LOCAL SYMMETRY AND FIBRATIONS

*Proof of Proposition 2.4.3.* It can be verified that  $\{\tilde{\mathcal{Y}}_1, \tilde{\mathcal{Y}}_2, \dots, \tilde{\mathcal{Y}}_M\}$  indeed forms a partition of  $\mathcal{Y}$ . Let us denote the transition rate matrix of  $Z$  by  $\tilde{A} = ((\tilde{a}_{i,j}))$ , where  $\tilde{a}_{i,j} = a_{f^{-1}(i), f^{-1}(j)}$ , and  $f^{-1}$  is the inverse of  $f$  in  $\text{Sym}(\mathcal{Y})$ . The proof will be complete if we show that the linear system  $\dot{z} = z\tilde{A}$  is lumpable. Pick  $\tilde{\mathcal{Y}}_i$ , and  $\tilde{\mathcal{Y}}_j$  for  $i \neq j$ , and let  $u, v \in \tilde{\mathcal{Y}}_i$  be arbitrarily chosen. See that  $u \in \tilde{\mathcal{Y}}_i$  implies  $f^{-1}(u) \in \mathcal{Y}_i$ . Then,

$$\sum_{l \in \tilde{\mathcal{Y}}_j} \tilde{a}_{u,l} = \sum_{l \in \tilde{\mathcal{Y}}_j} a_{f^{-1}(u), f^{-1}(l)} = \sum_{l \in \mathcal{X}_j} a_{s,l} = \sum_{l \in \mathcal{X}_j} a_{t,l} = \sum_{l \in \tilde{\mathcal{Y}}_j} a_{f^{-1}(v), f^{-1}(l)} = \sum_{l \in \tilde{\mathcal{Y}}_j} \tilde{a}_{v,l},$$

where  $s = f^{-1}(u), t = f^{-1}(v) \in \mathcal{X}_i$  and the equality for  $s$  and  $t$  holds by virtue of the lumpability of  $Y$ . This verifies the Dynkin's criterion for  $\dot{z} = z\tilde{A}$ .  $\square$

*Proof of Proposition 9.4.1.* Let us first assume  $x \in \text{fibre}(y)$ . In order to prove the vertices  $x, y$  are locally symmetric, we construct an isomorphism  $g : N_1(x) \rightarrow N_1(y)$  between  $G[N_1(x)]$  and  $G[N_1(y)]$  as follows

$$g(a) := s_G f_e^{-1}(a, x), \quad \forall a \in N_1(x). \quad (7.1.1)$$

Indeed,  $f_e^{-1}(a, x)$  is an edge in  $G[N_1(y)]$ , and therefore,  $g(a) \in N_1(y)$ . In order to check whether  $g$  is indeed an isomorphism, pick two vertices  $a, b \in N_1(x)$  such that  $(a, b) \in E$ . If  $b = x$ , the assertion follows straightforwardly. Therefore, we consider  $b \neq x$ . Then,  $(a, b) \in E$  implies the vertices  $a, b$ , and  $x$  form a triangle (see Figure 9.1).

Since  $f$  is a fibration,  $(f_v, f_e^{-1})$  is also a morphism because  $f_v$  and  $f_e^{-1}$  also commute with the source and target maps of  $G$ , i.e.,  $s_G f_e^{-1} = f_v s_G$  and  $t_G f_e^{-1} = f_v t_G$ . Now, let us consider the edge  $(a, b)$  in  $G[N_1(x)]$ . Since  $f$  is a fibration, there exists a unique edge  $f_e^{-1}(a, b) = (c, d) \in E$  such that  $f_e(c, d) = (a, b)$ , where  $d \in \text{fibre}(b)$ . Then,

$$(c, d) = (s_G f_e^{-1}(a, b), t_G f_e^{-1}(a, b)) = (f_v(a), f_v(b)) = (s_G f_e^{-1}(a, x), t_G f_e^{-1}(b, x)).$$

Therefore,  $g$  is indeed an isomorphism between  $G[N_1(x)]$  and  $G[N_1(y)]$  proving  $x \stackrel{1}{\sim} y$ .

Now, we prove the second part of the proposition. Let us assume  $x \stackrel{1}{\sim} y$ . In order to define a fibration  $f = (f_v, f_e)$ , let us first pick representatives for the equivalence classes of  $\stackrel{1}{\sim}$ . Let the injective map  $r : V \rightarrow V$  define the representatives, that is, for each  $x \in V$ , we have  $\langle x \rangle_1 = \langle r(x) \rangle_1$ . Then, consider the following maps

$$f_v(x) := r(x), \forall x \in V, \text{ and } f_e(a, b) = (g(a), g(b)),$$

where  $g \in \Psi_1$  is such that  $g(b) = r(b)$ . Please note that the choice of  $g$  depends on  $(a, b)$ . The epimorphism  $f$  defined above is indeed a fibration (Boldi and Vigna 2002).  $\square$

*Monotonicity fails for P-lifting*

It is intuitive that the monotonic decrease of KL divergence for finer partitions should carry over to lifting by the transition matrix. However, this is not the case as the following counterexample shows. Consider a transition probability matrix

$$T = \begin{pmatrix} 0.10 & 0.10 & 0.07 & 0.16 & 0.13 & 0.20 & 0.04 & 0.20 \\ 0.11 & 0.17 & 0.10 & 0.15 & 0.12 & 0.13 & 0.14 & 0.08 \\ 0.07 & 0.13 & 0.10 & 0.14 & 0.09 & 0.02 & 0.41 & 0.04 \\ 0.16 & 0.08 & 0.02 & 0.17 & 0.05 & 0.23 & 0.06 & 0.23 \\ 0.07 & 0.12 & 0.20 & 0.17 & 0.22 & 0.21 & 0.01 & 0.00 \\ 0.07 & 0.15 & 0.25 & 0.10 & 0.18 & 0.03 & 0.21 & 0.01 \\ 0.14 & 0.07 & 0.20 & 0.14 & 0.10 & 0.10 & 0.07 & 0.18 \\ 0.10 & 0.19 & 0.07 & 0.22 & 0.11 & 0.03 & 0.14 & 0.14 \end{pmatrix}.$$

Now, consider two partitions  $\{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}\}$  and  $\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$ . Clearly the latter partition is a refinement of the first. However, when we use  $P$ -lifting, the first partition yields a KL divergence of 0.0019067, while the second partition yields a higher KL divergence of 0.0308801.

## H.1 STATE-SPACE REDUCTION

Since there is no *perfect* graph<sup>1</sup> (see Behzad and Chartrand (1967) for a proof), we are certain that  $|H_G| < M$ . However, for  $M \geq 2$ , we can construct a unique *quasi-perfect* graph<sup>2</sup> (unique up to isomorphism) that is connected and entails  $|H_G| = M - 1$ , worst case scenario. We derive conditions for the mapping  $A$  to actually reduce state space. Before presenting our result in this context, let us define some necessary quantities.

Define  $R := (n_k : k \in H_G)$  and  $\mathcal{C} := \{C \in M_0^{|\mathcal{X}|} : \sum_{x \in \mathcal{X}: x_1=1} c(x) = 1, \sum_{x \in \mathcal{X}: x_1=0} c(x) = M - 1\}$ , where  $M_0 := \{0, 1, 2, \dots, M\}$ . Given  $R$  and a  $C \in \mathcal{C}$ , define the function  $F : (0, 1)^{|H_G| \times 2^n} \rightarrow \mathbb{R}$  as

$$F(x, y) := \left( \prod_{i \in H_G} x_i^{-n_i} \right) \left( \prod_{j \in \mathcal{X}} y_j^{-c(j)} \right) \left( \prod_{i \in H_G, j \in \mathcal{X}} \frac{1}{1 - x_i y_j} \right), \quad (8.1.1)$$

where  $x = (x_i : i \in H_G) \in (0, 1)^{|H_G|}$  and  $y = (y_j : j \in \mathcal{X}) \in (0, 1)^{2^n}$ . Also define its minimum on the open ball  $(0, 1)^{|H_G| \times 2^n}$  as follows

$$\chi(R, C) := \min_{x_i, y_j \in (0, 1) \forall i \in H_G, j \in \mathcal{X}} F(x, y). \quad (8.1.2)$$

Now we present our result regarding state space reduction.

**Result H.1.1.** *For  $\mathcal{G} \in \mathcal{G}_M$ , a necessary condition for the aggregation mapping  $A$  to engender state space reduction is*

$$M 2^{(M-1)(n-1)} \geq \binom{M-2+2^{n-1}}{M-1} \min_{C \in \mathcal{C}} M^{-a_0(|H_G|+2^n)} \chi(R, C), \quad (8.1.3)$$

for an absolute constant  $a_0 > 0$ . The following gives us a sufficient condition,

$$M 2^{(M-1)(n-1)} \geq \binom{M-2+2^{n-1}}{M-1} \max_{C \in \mathcal{C}} \chi(R, C). \quad (8.1.4)$$

*Proof of Result H.1.1.* Note that the aggregation mapping  $A$  is a measurable map from  $(\Omega, \mathcal{A})$  to  $(Y, \mathcal{M})$ , where

$$Y := \{v \in M_0^{|\mathcal{X}| \times H_G} : \sum_{k \in H_G} \sum_{x \in \mathcal{X}: x_1=1} v^k(x) = 1, \sum_{k \in H_G} \sum_{x \in \mathcal{X}: x_1=0} v^k(x) = M - 1, \sum_{x \in \mathcal{X}} v^k(x) = n_k\},$$

<sup>1</sup> A graph with two or more nodes is called perfect if for each pair of distinct vertices  $u$  and  $v$ ,  $d_u \neq d_v$ , i.e., no two vertices have the same degree.

<sup>2</sup> A graph with at least two nodes is quasi-perfect if there are precisely two vertices with the same degree

$n_k$  is the number of peers of degree  $k$  and  $\mathcal{M}$  is the  $\sigma$ -field generated by all subsets of  $Y$ .

Given  $\mathcal{G}$ , we try to find the size of  $Y$ . Suppose  $v \in Y$ . Elements of  $v$  must satisfy three sets of constraints, *viz.*,

$$\begin{aligned} \sum_{x \in \mathcal{X}} v^k(x) &= n_k \quad \forall k \in H_G, \\ \sum_{k \in H_G} \sum_{x \in \mathcal{X}: x_1=1} v^k(x) &= 1, \\ \sum_{k \in H_G} \sum_{x \in \mathcal{X}: x_1=0} v^k(x) &= M - 1. \end{aligned}$$

We treat this as a combinatorial problem of finding the number of contingency tables of non-negative elements, satisfying given row and column sums. In this context, regard the first set of equations as row constraints. These are fixed, given  $\mathcal{G}$ . Now set column constraints as

$$\sum_{k \in H_G} v^k(x) = c(x) \quad \forall x \in \mathcal{X},$$

where the column constraints are further constrained as follows

$$\sum_{x \in \mathcal{X}: x_1=1} c(x) = 1, \quad \sum_{x \in \mathcal{X}: x_1=0} c(x) = M - 1. \quad (8.1.5)$$

Let  $R := (n_k : k \in H_G)$ , and  $C := (c(x) : x \in \mathcal{X}) \in \mathcal{C}$ . Notice that  $R1 = C1 = M$ , the number of peers. Recall the definition of  $F$  and  $\chi$  from (8.1.1), (8.1.2) respectively.

Elements of the vector  $C$  can be partitioned into two equal halves. Each half can be thought of as a solution in non-negative integers to a linear Diophantine equation (the constraints in the definition of  $Y$ ). The first constraint is

$$\sum_{x \in \mathcal{X}: x_1=1} c(x) = 1,$$

which allows  $2^{n-1}$  solutions in non-negative integers. The second constraint is,

$$\sum_{x \in \mathcal{X}: x_1=0} c(x) = M - 1.$$

The above has  $\binom{M-1+2^{n-1}-1}{2^{n-1}-1} = \binom{M-2+2^{n-1}}{M-1}$  solutions in non-negative integers. Since the above two equations can be solved independently, the total number of admissible  $C$  is, therefore,  $|\mathcal{C}| = 2^{n-1} \binom{M-2+2^{n-1}}{M-1}$ .

Fix a  $C \in \mathcal{C}$ . Let  $\#(R, C)$  denote the number of  $|H_G| \times 2^n$  matrices (contingency tables) with non-negative elements satisfying row sum  $R$  and column sum  $C$ . Then, following Barvinok (2009), we get

$$\chi(R, C) \geq \#(R, C) \geq M^{-a_0(|H_G|+2^n)} \chi(R, C), \quad (8.1.6)$$

for an absolute constant  $a_0 > 0$ . Please refer to Barvinok (2009) for proof. Since any  $C \in \mathcal{C}$  is a valid choice for  $Y$ , we must have

$$|Y| = \sum_{C \in \mathcal{C}} \#(R, C) \geq |\mathcal{C}| \min_{C \in \mathcal{C}} M^{-a_0(|H_G|+2^n)} \chi(R, C).$$



Similarly, we get an upper bound as follows

$$|Y| \leq |\mathcal{C}| \max_{C \in \mathcal{C}} \chi(R, C).$$

Combining the above two, we get

$$|\mathcal{C}| \min_{C \in \mathcal{C}} M^{-a_0(|H_G|+2^n)} \chi(R, C) \leq |Y| \leq |\mathcal{C}| \max_{C \in \mathcal{C}} \chi(R, C). \quad (8.1.7)$$

Now, see that  $|\Omega| = M2^{M(n-1)}$ . We seek to find  $n \in \mathbb{N}$  such that  $|\Omega| \geq |Y|$ .

NECESSARY CONDITION

$$\begin{aligned} |\Omega| &\geq |Y| \\ \implies |\Omega| &\geq |\mathcal{C}| \min_{C \in \mathcal{C}} M^{-a_0(|H_G|+2^n)} \chi(R, C) \\ \iff M2^{M(n-1)} &\geq 2^{n-1} \binom{M-2+2^{n-1}}{M-1} \min_{C \in \mathcal{C}} M^{-a_0(|H_G|+2^n)} \chi(R, C) \\ \iff M2^{(M-1)(n-1)} &\geq \binom{M-2+2^{n-1}}{M-1} \min_{C \in \mathcal{C}} M^{-a_0(|H_G|+2^n)} \chi(R, C). \end{aligned}$$

SUFFICIENT CONDITION Set

$$\begin{aligned} |\Omega| &\geq \max_{C \in \mathcal{C}} \chi(R, C) \\ \iff M2^{M(n-1)} &\geq 2^{n-1} \binom{M-2+2^{n-1}}{M-1} \max_{C \in \mathcal{C}} \chi(R, C) \\ \iff M2^{(M-1)(n-1)} &\geq \binom{M-2+2^{n-1}}{M-1} \max_{C \in \mathcal{C}} \chi(R, C). \end{aligned}$$

□

## H.2 MEAN-FIELD THEORETIC ANALYSIS

*Proof of Lemma 10.2.1.* Fix  $u \in \mathcal{X}, k \in \mathbb{N}$ . It follows,

$$\begin{aligned} \sum_{Z \in \mathbb{N}_0^{|\mathcal{X}| \times \mathbb{N}}} z_u^{(k)} \sum_{Y: \sum_{\substack{Sv=u \\ \forall u, v \in \mathcal{X}, l \in \mathbb{N}}} y_v^{(l)} = z_u^{(l)}} P(Y) &= \sum_{z_u^{(k)}} \sum_{y_v^{(k)}: \sum_{Sv=u} y_v^{(k)} = z_u^{(k)}} \left( \sum_{Sv=u} y_v^{(k)} \right) P(\{y_v^{(k)} \mid Sv = u\}) \\ &= \sum_{Sv=u} E[z_v^{(k)}] \end{aligned}$$

□

*Proof of Result 10.2.1.* From (10.2.2) and applying Lemma 10.2.1, we get

$$\begin{aligned}
\frac{d}{dt} \mathbb{E}[Z] &= -\mathbb{E}[Z] + \mathbb{E}[Y] \\
&+ \sum_{Z \in \mathbb{N}_0^{|\mathcal{X}| \times \mathbb{N}}} Z \sum_{\substack{Y: \sum_{Sv=u} y_v^{(l)} = z_u^{(l)} \\ \forall u, v \in \mathcal{X}, l \in \mathbb{N}}} \left[ \sum_{l \in \mathbb{N}, u \in \mathcal{X}, i \in [n]} \gamma_{l,u,i}(Y - \varrho(l, u, i)) \mathbb{P}(Y - \varrho(l, u, i)) - \right. \\
&\quad \left. \sum_{l \in \mathbb{N}, u \in \mathcal{X}, i \in [n]} \gamma_{l,u,i}(Y) \mathbb{P}(Y) \right] \\
&= -\mathbb{E}[Z] + \mathbb{E}[Y] \\
&+ \sum_{l \in \mathbb{N}, u \in \mathcal{X}, i \in [n]} \sum_{Z \in \mathbb{N}_0^{|\mathcal{X}| \times \mathbb{N}}} \sum_{\substack{Y: \sum_{Sv=u} y_v^{(l)} = z_u^{(l)} \\ \forall u, v \in \mathcal{X}, l \in \mathbb{N}}} \varrho(l, u, i) \gamma_{l,u,i}(Y - \varrho(l, u, i)) \mathbb{P}(Y - \varrho(l, u, i)) \\
&= -\mathbb{E}[Z] + \mathbb{E}[Y] + \sum_{l \in \mathbb{N}, u \in \mathcal{X}, i \in [n]} \varrho(l, u, i) \mathbb{E}[\gamma_{l,u,i}(Y)]
\end{aligned}$$

The second line is arrived at by addition and subtraction of  $\varrho(l, u, i)$  and rearrangement of summands.  $\square$

*Proof of Lemma 10.2.2.* 1. Notice that, for  $u, v \in \mathcal{X} : Sv = u$  and  $u_{i+1} = 1 \implies v_i = 1$ . Therefore,

$$\sum_{u \in \mathcal{X}: u_{i+1}=1} \sum_{v \in \mathcal{X}: Sv=u} w_v^{(k)} = \sum_{v \in \mathcal{X}: v_i=1} w_v^{(k)} = p_k(i).$$

2. We simplify the left hand side and omit terms whenever they turn out to be 0.

$$\begin{aligned}
&\sum_{u \in \mathcal{X}: u_i=1} \sum_{j \in [n]} (\lambda^{(k)}(u - e_j, u) - \lambda^{(k)}(u, u + e_j)) \\
&= \sum_{u \in \mathcal{X}: u_i=1} (\lambda^{(k)}(u - e_i, u) - \lambda^{(k)}(u, u + e_i)) \\
&\quad + \sum_{u \in \mathcal{X}: u_i=1} \sum_{j \in [n] \setminus \{i\}} (\lambda^{(k)}(u - e_j, u) - \lambda^{(k)}(u, u + e_j)) \\
&= \sum_{u \in \mathcal{X}: u_i=1} \lambda^{(k)}(u - e_i, u) + \sum_{j \in [n] \setminus \{i\}} \sum_{u \in \mathcal{X}: u_i=1} (\lambda^{(k)}(u - e_j, u) - \lambda^{(k)}(u, u + e_j)) \\
&= \sum_{u \in \mathcal{X}: u_i=1} \lambda^{(k)}(u - e_i, u).
\end{aligned}$$

This is because,

$$\begin{aligned}
& \sum_{j \in [n] \setminus \{i\}} \sum_{u \in \mathcal{X}: u_i=1} (\lambda^{(k)}(u - e_j, u) - \lambda^{(k)}(u, u + e_j)) \\
&= \sum_{j \in [n] \setminus \{i\}} \left[ \sum_{u \in \mathcal{X}: u_i=1, u_j=1} (\lambda^{(k)}(u - e_j, u) - \lambda^{(k)}(u, u + e_j)) \right. \\
&\quad \left. + \sum_{u \in \mathcal{X}: u_i=1, u_j=0} (\lambda^{(k)}(u - e_j, u) - \lambda^{(k)}(u, u + e_j)) \right] \\
&= \sum_{j \in [n] \setminus \{i\}} \left[ \sum_{u \in \mathcal{X}: u_i=1, u_j=1} \lambda^{(k)}(u - e_j, u) - \sum_{u \in \mathcal{X}: u_i=1, u_j=0} \lambda^{(k)}(u, u + e_j) \right] \\
&= \sum_{j \in [n] \setminus \{i\}} \left[ \sum_{u \in \mathcal{X}: u_i=1} \left( \lambda^{(k)}(u - e_j, u) - \lambda^{(k)}(u - e_j, u) \right) \right] \\
&= 0.
\end{aligned}$$

Such a rearrangement of summands is possible because  $u \in \mathcal{X} : u_i = 1, u_j = 1 \implies v = u - e_j \in \mathcal{X} : v_i = 1, v_j = 0$ . This completes the proof.  $\square$

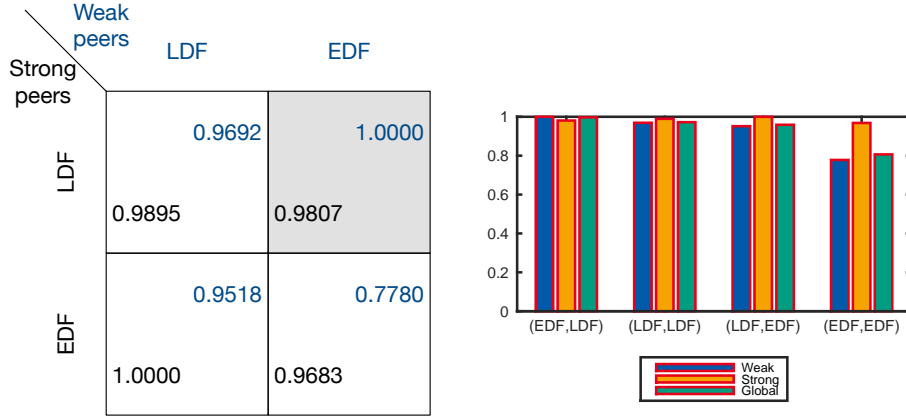
*Proof of Result 10.2.2.* Summing both sides of (10.2.6) and using Lemma 10.2.2, we get

$$\begin{aligned}
\sum_{u \in \mathcal{X}: u_{i+1}=1} w_u^{(k)} &= \sum_{u \in \mathcal{X}: u_{i+1}=1} \sum_{v \in \mathcal{X}: Sv=u} w_v^{(k)} \\
&\quad + \sum_{u \in \mathcal{X}: u_{i+1}=1} \sum_{v \in \mathcal{X}: Sv=u} \sum_{j \in [n]} \left( \lambda^{(k)}(v - e_j, v) - \lambda^{(k)}(v, v + e_j) \right) \\
\implies p_k(i+1) &= p_k(i) + \sum_{u \in \mathcal{X}: u_i=1} \lambda^{(k)}(u - e_i, u).
\end{aligned}$$

Summing the above according to (10.2.8), we get the other recurrence relation pertaining to global performance. The fact that buffer probabilities are non-decreasing in buffer indices follows from the non-negativity of  $\lambda^{(k)}$ 's.  $\square$

### H.3 GAME THEORETIC ARGUMENT

In Section 10.2, we showed that SCHEDMix could guarantee good playback continuity at a smaller start-up latency and smaller unsuccessful download rate. The question, however, remains why the strong peers should opt to play LDF. To answer this question, we bring in a game theoretic perspective by pitting weak peers against strong ones within the purview of the simplified two-degree system discussed in Section 10.2.3. Let  $\mathcal{S}_1 = \mathcal{S}_2 = \{\text{LDF}, \text{EDF}\}$  be the strategy profiles of weak and strong peers, respectively and  $\mathcal{S} := \mathcal{S}_1 \times \mathcal{S}_2$  denote the set of all possible strategy vectors. We claim that  $(\text{EDF}, \text{LDF}) \in \mathcal{S}$  is a Nash equilibrium (Nisan et al. 2007), choosing the playback continuity probabilities as respective utility functions.



**Figure H.1:** Comparison of all possible strategy vectors in  $\mathcal{S}$ . The utility function under consideration is the playback continuity probability. The weak peers benefit more from SCHEDMIX.

Let us first argue about the weak peers' strategy against LDF-playing strong peers. From the perspective of weak peers, LDF is not a "rational strategy", because under the pure LDF strategy, the weak remains weak (see (10.2.18)). On the other hand, weak peers can eventually outperform the LDF-playing strong peers if they are greedy (see Figure 10.2). From the perspective of strong peers, they are better-off playing LDF against EDF-playing weak peers, establishing that (EDF, LDF) is a "Nash equilibrium". For an illustration, refer to Figure H.1 and verify that (EDF, LDF)  $\in \mathcal{S}$  is indeed a Nash equilibrium.

Interestingly (LDF, EDF) is also *Nash*. This explains the "boon of heterogeneity" phenomenon that we mentioned earlier. However, this is suboptimal (see Figure H.1). When the strong peers play LDF, they act as *pseudo-sources* and facilitate propagation of rarest chunks. That is why (EDF, LDF) is more beneficial from the perspective of global performance. It must also be noted that the utility functions depend on different choices of  $n, k_1, k_2, \pi_1 = 1 - \pi_2, M$ , therefore, Figure H.1 should only be deemed as an illustration of the game theoretic argument in a realistic set-up with moderate buffer size. Extreme parameter choices, e.g.,  $n \rightarrow \infty, \frac{\pi_1}{\pi_2} \rightarrow \infty$  are excluded from consideration.

## NOTATION

SYMBOL	DESCRIPTION
$\mathbb{N}$	The set of natural numbers.
$\mathbb{R}$	The set of real numbers.
$[N]$	The set $\{1, 2, 3, \dots, N\}$ for $N \in \mathbb{N}$ .
$\mathbb{N}_0$	The set of nonnegative whole numbers.
$\mathbb{R}_+$	The set of positive real numbers.
$\mathcal{B}(R)$	The $\sigma$ -field of Borel subsets of $R \subseteq \mathbb{R}$ .
$\text{Int } F$	The interior of a set $F \in \mathcal{B}(\mathbb{R}^N)$ .
$\text{Cl } F$	The closure of a set $F \in \mathcal{B}(\mathbb{R}^N)$ .
$\text{Bnd } F$	The boundary of a set $F \in \mathcal{B}(\mathbb{R}^N)$ .
$\mathcal{D}f$	The effective domain of an extended real-valued function $f$ , i.e., $\mathcal{D}f := \{x \in \mathbb{R} \mid f(x) < \infty\}$ .
$\text{ess sup}$	The essential supremum.
$ A $	The cardinality of a set $A$ .
$2^A$	The class of all subsets of a set $A$ .
$\Lambda(N, K)$	The set of all non-negative integer solutions to the Diophantine equation $x_1 + x_2 + \dots + x_K = N$ , i.e., $\Lambda(N, K) := \{x = (x_1, x_2, \dots, x_K) \in \mathbb{N}_0^K \mid x_1 + x_2 + \dots + x_K = N\}$ , for $N, K \in \mathbb{N}$ .
$\text{Sym}(A)$	The symmetric group on a set $A$ .
$\text{Aut}(G)$	The automorphism group of a graph $G$ .
$\text{per } A$	The permanent of a matrix $A$ .
$\mathcal{T}$	The time interval $[0, T]$ .
$D = D(\mathcal{T})$	The Polish space of real functions $f$ on $\mathcal{T}$ that are right continuous and have left hand limits.
$\mathbb{1}(A)$	The indicator (or characteristic) function of a set $A$ .
$(a)_b$	The quantity $a(a-1)(a-2) \dots (a-b+1)$ for $a > b$ and $a, b \in \mathbb{N}$ .
$\langle Z \rangle$	The predictable quadratic variation of the process $Z$ .
$[Z]$	The optional quadratic variation of the process $Z$ .
$\xrightarrow{\text{P}}$	Convergence in probability.
$\xrightarrow{\text{a.s.}}$	Almost sure convergence.
$\xrightarrow{\mathcal{D}}$	Weak convergence.
$O()$	Big o notation.
$o()$	Small o notation.



## ACRONYMS

---

ABM	Agent-based Model
ADMM	Alternating Direction Method of Multipliers
BA	Barabási-Albert
BCS	Bioinspired Communication Systems
CBQA	Cost-Based Queue-Aware
CCDF	Complementary Cumulative Distribution Function
CDF	Cumulative Distribution Function
CDN	Content Distribution Network
CIM	Conditional Intensity Matrix
CLT	Central Limit Theorem
CM	Configuration Model
CME	Chemical Master Equation
CRC	Collaborative Research Centre
CRN	Chemical Reaction Network
CTBN	Continuous Time Bayesian Network
CTMC	Continuous Time Markov Chain
DCFTP	Dominated Coupling From The Past
DFG	German Research Foundation
DTMC	Discrete Time Markov Chain
ECMP	Equal-cost Multi-path routing
EDF	Earliest Deadline First
ER	Erdős-Rényi
ESI	Enzyme-Substrate-Inhibitor
FCFS	First Come First Served
FCLT	Functional Central Limit Theorem
FIFO	First In First Out

FJ	Fork-Join
GBP	General Branching Process
ID	Information-Dissemination
iid	independent and identically distributed
IoT	Internet of Things
IPS	Interacting Particle System
IT	Information Technology
JIQ	Join-Idle-Queue
JMC	Join-Minimum-Cost
JSQ	Join-Shortest-Queue
KL	Kullback-Leibler
LDF	Latest Deadline First
LDP	Large Deviations Principle
LLN	Law of Large Numbers
LNA	Linear Noise Approximation
MABM	Markovian Agent-based Model
MAKI	Multi-Mechanism Adaptation for the Future Internet
MAPK	Mitogen-activated Protein Kinase
MDS	Maximum Distance Separable
MGF	Moment Generating Function
MM	Michaelis-Menten
MPI	Message Passing Interface
Multi-path TCP	Multi-path Transmission Control Protocol
ODE	Ordinary Differential Equation
P2P	Peer-to-Peer
PDF	Probability Density Function
PGF	Probability Generating Function
PGM	Probabilistic Graphical Model
PMF	Probability Mass Function



psd	positive semi-definite
PT	Poisson-type
QSSA	Quasi-Steady State Approximation
rQSSA	reversible QSSA
SAN	Stochastic Automata Network
SEIR	Susceptible-Exposed-Infected-Recovered
SI	Susceptible-Infected
SIR	Susceptible-Infected-Recovered
SIS	Susceptible-Infected-Susceptible
sQSSA	standard QSSA
SRPT	Shortest Remaining Processing Time
ssLNA	Slow-scale Linear Noise Approximation
TCP	Transmission Control Protocol
tQSSA	total QSSA
WS	Watts-Strogatz
whp	with high probability



## BIBLIOGRAPHY

---

- Amazon Elastic Compute Cloud EC2* (2016). URL: <https://aws.amazon.com/ec2/> (visited on 07/07/2016) (cit. on p. 23).
- Amazon.com, Inc. (Dec. 11, 2017). *Amazon Web Services (AWS)*. Accessed: 11-12-2017 (cit. on p. 34).
- Andersen, P. K., Ø. Borgan, R. D. Gill, and N. Keiding (1993). *Statistical models based on counting processes*. Springer Series in Statistics. Springer-Verlag, New York, pp. xii+767 (cit. on pp. 101, 106–108).
- Anderson, D. F., D. Cappelletti, M. Koyama, and T. G. Kurtz (Aug. 2017). “Non-explosivity of stochastically modeled reaction networks that are complex balanced”. In: *ArXiv e-prints* (cit. on p. 78).
- Anderson, D. F. and T. G. Kurtz (2011). “Continuous time Markov chain models for chemical reaction networks”. In: *Design and analysis of biomolecular circuits*. Springer, pp. 3–42 (cit. on pp. 13, 14, 85, 86).
- Anderson, W. J. (1991). *Continuous-Time Markov Chains*. Springer-Verlag New York (cit. on p. 20).
- Angluin, D. (1980). “Local and global properties in networks of processors (Extended Abstract)”. In: *Proceedings of the twelfth annual ACM symposium on Theory of computing (STOC)* (cit. on p. 136).
- Apache Spark* (2016). URL: <http://spark.apache.org/> (visited on 07/17/2016) (cit. on p. 23).
- Apers, S., F. Ticozzi, and A. Sarlette (2017). “Lifting Markov Chains To Mix Faster: Limits and Opportunities”. In: *arXiv preprint arXiv:1705.08253* (cit. on p. 137).
- Arazi, A., E. Ben-Jacob, and U. Yechiali (2004). “Bridging genetic networks and queueing theory”. In: *Physica A: Statistical Mechanics and its Applications* 332, pp. 585–616 (cit. on pp. 13, 16).
- Arvind, V., J. Köbler, G. Rattan, and O. Verbitsky (2016). “Graph Isomorphism, Color Refinement, and Compactness”. In: *Computational Complexity* 26.3, pp. 627–685 (cit. on pp. 136, 137).
- Ash, R. B. (1972). *Real Analysis and Probability*. Academic Press (cit. on pp. 23, 161, 162).
- Assaf, M. and B. Meerson (2017). “WKB theory of large deviations in stochastic populations”. In: *J. Phys. A* 50.26, p. 263001 (cit. on p. 78).
- Atkinson, K. E. (2008). *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge University Press (cit. on p. 41).
- Babai, L. (Dec. 2015). “Graph Isomorphism in Quasipolynomial Time”. In: *ArXiv e-prints* (cit. on p. 119).
- Babai, L., P. Erdős, and S. M. Selkow (1980). “Random Graph Isomorphism”. In: *SIAM Journal on Computing* 9.3, pp. 628–635 (cit. on p. 137).
- Bacelli, F., A. M. Makowski, and A. Shwartz (1989). “The Fork-Join Queue and Related Systems with Synchronization Constraints: Stochastic Ordering and Computable Bounds”. In: *Advances in Applied Probability*, pp. 629–660 (cit. on pp. 5, 13, 25, 30).

- Baker, C. T. H. (1977). *The Numerical Treatment of Integral Equations*. Oxford University Press (cit. on p. 41).
- Ball, K., T. G. Kurtz, L. Popovic, and G. A. Rempala (2006). "Asymptotic analysis of multiscale approximations to reaction networks". In: *Ann. Appl. Probab.* 16.4, pp. 1925–1961 (cit. on pp. 7, 79, 81, 96).
- Balsamo, S., L. Donatiello, and N. M. V. Dijk (1998). "Bound Performance Models of Heterogeneous Parallel Processing Systems". In: *IEEE Transactions on Parallel and Distributed Systems* 9.10, pp. 1041–1056 (cit. on p. 13).
- Banisch, S. (2016). *Markov Chain Aggregation for Agent-Based Models*. Springer International Publishing (cit. on pp. 1, 136).
- Bapat, R. B. and M. I. Beg (1989). "Order Statistics for Nonidentically Distributed Variables and Permanents". In: *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)* 51.1, pp. 79–93 (cit. on pp. 175, 180).
- Barabási, A.-L. and R. Albert (1999). "Emergence of Scaling in Random Networks". In: *Science* 286.5439, pp. 509–512 (cit. on p. 151).
- Barakat, H. M. and Y. H. Abdelkader (2004). "Computing the moments of order statistics from nonidentical random variables". In: *Statistical Methods and Applications* 13.1, pp. 15–26 (cit. on pp. 175, 177, 180).
- Barato, A. C. and H. Hinrichsen (2009). "Nonequilibrium phase transition in a spreading process on a timeline". In: *Journal of Statistical Mechanics* (cit. on p. 110).
- Barik, D., M. R. Paul, W. T. Baumann, Y. Cao, and J. J. Tyson (2008). "Stochastic simulation of enzyme-catalyzed reactions with disparate timescales". In: *Biophys. J.* 95.8, pp. 3563–3574 (cit. on p. 92).
- Barndorff-Nielsen, O. (1963). "On the limit behaviour of extreme order Statistics". In: *Ann. Math. Statist.* 34, pp. 992–1002 (cit. on p. 106).
- Baroni, E., R. van der Hofstad, and J. Komjathy (2015). "First passage percolation on random graphs with infinite variance degrees". In: <https://arxiv.org/pdf/1506.01255> (cit. on p. 110).
- Barvinok, A. (2009). "Asymptotic Estimates for the Number of Contingency Tables, Integer Flows, and Volumes of Transportation Polytopes". In: *International Mathematics Research Notices* 2009.2, pp. 348–385 (cit. on p. 200).
- Behzad, M. and G. Chartrand (1967). "No Graph is Perfect". In: *American Mathematical Monthly*, pp. 962–963 (cit. on p. 199).
- Berkholz, C., P. Bonsma, and M. Grohe (2013). "Tight Lower and Upper Bounds for the Complexity of Canonical Colour Refinement". In: *Algorithms – ESA 2013*, pp. 145–156 (cit. on p. 136).
- Bersani, A. M. and G. Dell'Acqua (2011). "Asymptotic expansions in enzyme reactions with high enzyme concentrations". In: *Math. Methods Appl. Sci.* 34.16, pp. 1954–1960 (cit. on p. 79).
- Bersani, A. M., M. G. Pedersen, E. Bersani, and F. Barcellona (2005). "A mathematical approach to the study of signal transduction pathways in MAPK cascade". In: *Ser. Adv. Math. Appl. Sci.* 69, p. 124 (cit. on p. 96).
- Biancalani, T. and M. Assaf (2015). "Genetic toggle switch in the absence of cooperative binding: exact results". In: *Phys. Rev. Lett.* 115 (20), p. 208101 (cit. on p. 78).

- Billingsley, P. (1999). *Convergence of probability measures*. Second. Wiley Series in Probability and Statistics: Probability and Statistics. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, pp. x+277 (cit. on pp. 75, 99).
- BitTorrent (2018). URL: <http://www.bittorrent.com/> (visited on 03/01/2018) (cit. on p. 3).
- Bolch, G., S. Greiner, H. de Meer, and K. S. Trivedi (2006). *Queueing Networks and Markov Chains* (cit. on p. 65).
- Boldi, P., V. Lonati, M. Santini, and S. Vigna (2006). "Graph fibrations, graph isomorphism, and PageRank". In: *RAIRO-Theoretical Informatics and Applications* 40.2, pp. 227–253 (cit. on p. 128).
- Boldi, P. and S. Vigna (2002). "Fibrations of graphs". In: *Discrete Mathematics* 243.1-3, pp. 21–66 (cit. on pp. 128, 129, 197).
- Borghans, J. A. M., R. J. De Boer, and L. A. Segel (1996). "Extending the quasi-steady state approximation by changing variables". In: *Bull. Math. Biol.* 58.1, pp. 43–63 (cit. on pp. 80, 88, 90, 188).
- Boucheron, S., G. Lugosi, and P. Massart (2013). *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press (cit. on p. 57).
- Boxma, O. J., G. Koole, and Z. Liu (1994). *Queueing-theoretic Solution Methods for Models of Parallel and Distributed Systems*. Centrum voor Wiskunde en Informatica, Department of Operations Research, Statistics, and System Theory (cit. on pp. 5, 13, 23, 161).
- Bressloff, P. C. (2017). "Stochastic switching in biology: from genotype to phenotype". In: *J. Phys. A* 50.13, p. 133001 (cit. on p. 78).
- Bressloff, P. C. and J. M. Newby (2013). "Metastability in a stochastic neural network modeled as a velocity jump Markov process". In: *SIAM J. Appl. Dyn. Syst.* 12.3, pp. 1394–1435 (cit. on p. 78).
- Briggs, G. E. and J. B. S. Haldane (1925). "A note on the kinetics of enzyme action". In: *Biochem. J.* 19.2, p. 338 (cit. on p. 80).
- Buchholz, P. (1994). "Exact and ordinary lumpability in finite Markov chains". In: *Journal of Applied Probability* 31.1, pp. 59–75 (cit. on pp. 17, 18).
- Buchholz, P. and P. Kemper (2004). "Kronecker based matrix representations for large Markov models". In: *Validation of Stochastic Systems*. Springer, pp. 256–295 (cit. on p. 138).
- Callaway, D. S., M. E. Newman, S. H. Strogatz, and D. J. Watts (2000). "Network robustness and fragility: Percolation on random graphs". In: *Physical Review Letters* 85.25, p. 5468 (cit. on p. 110).
- Chen, F., L. Lovász, and I. Pak (1999). "Lifting Markov Chains to Speed Up Mixing". In: *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*. STOC '99. Atlanta, Georgia, USA: ACM, pp. 275–281 (cit. on p. 137).
- Chen, Y., S. Alspaugh, and R. Katz (2012). "Interactive Analytical Processing in Big Data Systems: A Cross-industry Study of MapReduce Workloads". In: *Proceedings of the VLDB Endowment* 5.12, pp. 1802–1813 (cit. on pp. 33, 42, 45, 67).
- Choi, B. S., G. A. Rempała, and J. Kim (2017). "Beyond the Michaelis-Menten equation: Accurate and efficient estimation of enzyme kinetic parameters" (cit. on p. 97).
- Chong, J., N. Satish, B. Catanzaro, K. Ravindran, and K. Keutzer (2007). "Efficient Parallelization of H.264 Decoding with Macro Block Level Scheduling". In: *IEEE ICME*, pp. 1874–1877 (cit. on pp. 6, 25, 29).

- Cookson, N. A., W. H. Mather, T. Danino, O. Mondragón-Palomino, R. J. Williams, L. S. Tsimring, and J. Hasty (2011). "Queueing up for enzymatic processing: correlated signaling through coupled degradation". In: *Molecular systems biology* 7.1, p. 561 (cit. on pp. 16, 17).
- Cornish-Bowden, A. (2004). *Fundamentals of enzyme kinetics*. Portland Press (cit. on pp. 77, 79, 187).
- Courtois, P. J. (2014). *Decomposability: queueing and computer system applications*. Academic Press (cit. on p. 137).
- Darden, T. A. (1979). "A pseudo-steady-state approximation for stochastic chemical kinetics". In: *Rocky Mt. J. Math.* 9.1, pp. 51–71 (cit. on p. 85).
- Darden, T. A. (1982). "Enzyme kinetics: stochastic vs. deterministic models". In: *Instabilities, bifurcations, and fluctuations in chemical systems*. University of Texas Press, Austin, pp. 248–272 (cit. on p. 85).
- Dean, J. and S. Ghemawat (2008). "MapReduce: Simplified Data Processing on Large Clusters". In: *Communications of the ACM* 51.1, pp. 107–113 (cit. on pp. 4, 11).
- Decreusefond, L., J.-S. Dhersin, P. Moyal, and V. C. Tran (2012). "Large graph limit for an SIR process in random network with heterogeneous connectivity". In: *Ann. Appl. Probab.* 22.2, pp. 541–575 (cit. on p. 104).
- Dell'Acqua, G. and A. M. Bersani (2011). "Quasi-steady state approximations and multistability in the double phosphorylation-dephosphorylation cycle". In: *International joint conference on biomedical engineering systems and technologies*. Springer, pp. 155–172 (cit. on p. 96).
- Dell'Acqua, G. and A. M. Bersani (2012). "A perturbation solution of Michaelis–Menten kinetics in a "total" framework". In: *J Math Chem.* 50.5, pp. 1136–1148 (cit. on p. 81).
- Dembo, A. and O. Zeitouni (2010). *Large deviations techniques and applications*. Springer-Verlag Berlin Heidelberg (cit. on pp. 37, 38, 40, 172).
- Deng, K., P. G. Mehta, and S. P. Meyn (Dec. 2011). "Optimal Kullback-Leibler Aggregation via Spectral Theory of Markov Chains". In: *IEEE Transactions on Automatic Control* 56.12, pp. 2793–2808 (cit. on pp. 129, 130).
- Dingee, J. W. and A. B. Anton (2008). "A new perturbation solution to the Michaelis-Menten problem". In: *AIChE J.* 54.5, pp. 1344–1357 (cit. on p. 79).
- Duffield, N. G. (1994). "Exponential Bounds for Queues with Markovian Arrivals". In: *Queueing Systems* 17.3, pp. 413–430 (cit. on pp. 34, 39, 42, 173).
- Durrett, R. (2010a). *Probability: Theory and Examples*. Fourth. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, pp. x+428 (cit. on pp. 23, 38, 101, 173).
- Durrett, R. (2010b). *Random graph dynamics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, pp. x+210 (cit. on p. 142).
- Elbert Simões, J., D. R. Figueiredo, and V. C. Barbosa (May 2016). "Local symmetry in random graphs". In: *ArXiv e-prints* (cit. on pp. 119, 125, 126).
- Ethier, S. N. and T. G. Kurtz (1986). *Markov Processes: Characterization and Convergence*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. Characterization and convergence. John Wiley & Sons, Inc., New York, pp. x+534 (cit. on pp. 6, 14, 63, 72, 73, 75, 83).

- Facebook Live* (2018). URL: <https://live.fb.com/> (visited on 02/27/2018) (cit. on pp. 1, 53).
- Feret, J., T. Henzinger, H. Koepl, and T. Petrov (2012). “*Lumpability abstractions of rule-based systems*”. In: *Theoretical Computer Science* 431.Supplement C. Modelling and Analysis of Biological Systems, pp. 137–164 (cit. on p. 136).
- Fidler, M. and Y. Jiang (2016). “Non-Asymptotic Delay Bounds for (k,l) Fork-Join Systems and Multi-Stage Fork-Join Networks”. In: *Proceedings of of IEEE INFOCOM* (cit. on p. 13).
- Fleming, T. R. and D. P. Harrington (n.d.). *Counting processes and survival analysis* (cit. on p. 101).
- Ford, A., C. Raiciu, M. Handley, and O. Bonaventure (Jan. 2013). *TCP Extensions for Multipath Operation with Multiple Addresses*. RFC 6824. Internet Engineering Task Force (cit. on p. 4).
- Ford, A. and M. Scharf (2013). “*Multipath TCP (MPTCP) Application Interface Considerations*”. In: (cit. on p. 4).
- França, G. and J. Bento (2017). “Markov Chain Lifting and Distributed ADMM”. In: *IEEE Signal Processing Letters* 24.3, pp. 294–298 (cit. on p. 137).
- Frömmgen, A., B. Richerzhagen, J. Rückert, D. Hausheer, R. Steinmetz, and A. Buchmann (2015). “Towards the Description and Execution of Transitions in Networked Systems”. In: *AIMS* (cit. on pp. 3, 153).
- Frömmgen, A., A. Rizk, T. E. M. Weller, B. Koldehofe, A. Buchmann, and R. Steinmetz (2017). “A Programming Model for Application-defined Multipath TCP Scheduling”. In: *Proc. 18th ACM/IFIP/USENIX Middleware Conf.* Pp. 134–146 (cit. on p. 48).
- Gadgil, C., C. H. Lee, and H. G. Othmer (2005). “*A stochastic analysis of first-order reaction networks*”. In: *Bulletin of Mathematical Biology* 67.5, pp. 901–946 (cit. on pp. 13, 15).
- Ganesh, A. J., N. O’Connell, and D. J. Wischik (2004). *Big Queues*. Springer-Verlag Berlin Heidelberg (cit. on p. 38).
- Ganguly, A., T. Petrov, and H. Koepl (Sept. 2014). “*Markov chain aggregation and its applications to combinatorial reaction networks*”. In: *Journal of Mathematical Biology* 69.3, pp. 767–797 (cit. on p. 19).
- Geiger, B. C., T. Petrov, G. Kubin, and H. Koepl (Apr. 2015). “*Optimal Kullback-Leibler Aggregation via Information Bottleneck*”. In: *IEEE Transactions on Automatic Control* 60.4, pp. 1010–1022 (cit. on pp. 129, 130, 133).
- Gillespie, D. T. (1977). “Exact stochastic simulation of coupled chemical reactions”. In: *J. Phys. Chem.* 81.25, pp. 2340–2361 (cit. on pp. 86, 90).
- Godsil, C. and G. F. Royle (2013). *Algebraic Graph Theory*. Vol. 207. Springer-Verlag New York (cit. on pp. 119, 122, 126).
- Gómez-Urbe, C. A., G. C. Verghese, and L. A. Mirny (2007). “Operating regimes of signaling cycles: statics, dynamics, and noise filtering”. In: *PLoS Comput. Biol.* 3.12, e246 (cit. on p. 96).
- Grima, R., D. R. Schmidt, and T. J. Newman (2012). “*Steady-state fluctuations of a genetic feedback loop: An exact solution*”. In: *J. Chem. Phys.* 137.3, p. 035104 (cit. on p. 97).
- Hammes, G. (2012). *Enzyme catalysis and regulation*. Elsevier (cit. on pp. 77, 79).
- Harris, T. E. (1963). *The Theory of Branching Processes*. Springer-Verlag Berlin Heidelberg (cit. on pp. 39, 171, 173).



- Hashem, I. A. T., N. B. Anuar, A. Gani, I. Yaqoob, F. Xia, and S. U. Khan (2016). "MapReduce: Review and Open Challenges". In: *Scientometrics*, pp. 1–34 (cit. on pp. 4, 11).
- Heffes, H. and D. Lucantoni (Sept. 1986). "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance". In: *IEEE Journal on Selected Areas in Communications* 4.6, pp. 856–868 (cit. on pp. 33, 42, 45, 67).
- Helland, I. S. (1982). "Central limit theorems for martingales with discrete or continuous time". In: *Scand. J. Statist.* 9.2, pp. 79–94 (cit. on p. 104).
- Hemberg, M. and M. Barahona (2008). "A Dominated Coupling From The Past algorithm for the stochastic simulation of networks of biochemical reactions". In: *BMC Systems Biology* 2.1, p. 42 (cit. on p. 137).
- Hendrickx, J. M. (July 2014). "Views in a Graph: To Which Depth Must Equality Be Checked?" In: *IEEE Transactions on Parallel and Distributed Systems* 25.7, pp. 1907–1912 (cit. on p. 126).
- Hinrichsen, H. (2006). "Non-equilibrium phase transitions". In: *Phys. A* 369.1, pp. 1–28 (cit. on p. 110).
- Hochendoner, P., C. Ogle, and W. H. Mather (2014). "A queueing approach to multi-site enzyme kinetics". In: *Interface focus* 4.3, p. 20130077 (cit. on pp. 16, 17).
- Hofstad, R. van der (2010). "Percolation and random graphs". In: *New perspectives in stochastic geometry*. Oxford Univ. Press, Oxford, pp. 173–247 (cit. on p. 110).
- Hofstad, R. van der (2017). *Random graphs and complex networks*. Vol. 1. Cambridge Series in Statistical and Probabilistic Mathematics (cit. on p. 99).
- Hopps, C. E. (2000). "Analysis of an equal-cost multi-path algorithm". In: (cit. on p. 4).
- Howe, J. (2006). "The rise of crowdsourcing". In: *Wired magazine* 14.6, pp. 1–4 (cit. on p. 2).
- Iscoe, I., P. Ney, and E. Nummelin (1985). "Large Deviations of Uniformly Recurrent Markov Additive Processes". In: *Advances in Applied Mathematics* 6.4, pp. 373–412 (cit. on pp. 33, 34, 37–39, 42, 171, 173).
- Jacobsen, K. A., M. G. Burch, J. H. Tien, and G. A. Rempała (2016). "The large graph limit of a stochastic epidemic model on a dynamic multilayer network". In: *arXiv preprint arXiv:1605.02809* (cit. on pp. 101–105, 109, 115, 116, 194, 195).
- Janson, S. (2009). "The probability that a random multigraph is simple". In: *Combinatorics, Probability and Computing* 18.1-2 (cit. on p. 100).
- Janson, S. and M. J. Luczak (2009). "A new approach to the giant component problem". In: *Random Structures Algorithms* 34.2, pp. 197–216 (cit. on p. 113).
- Janson, S., M. Luczak, and P. Windridge (2014). "Law of large numbers for the SIR epidemic on a random graph with given degrees". In: *Random Structures Algorithms* 45.4. Paging previously given as: 724–761, pp. 726–763 (cit. on pp. 104, 114).
- Joshi, G., E. Soljanin, and G. Wornell (Apr. 2017). "Efficient Redundancy Techniques for Latency Reduction in Cloud Systems". In: *ACM Trans. Model. Perform. Eval. Comput. Syst.* 2.2, 12:1–12:30 (cit. on pp. 5, 50, 58).
- Kandula, S., S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken (2009). "The Nature of Data Center Traffic: Measurements & Analysis". In: *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*. Chicago, Illinois, USA: ACM, pp. 202–208 (cit. on pp. 33, 42, 45, 67).



- Kang, H.-W. and T. G. Kurtz (2013). "Separation of time-scales and model reduction for stochastic reaction networks". In: *Ann. Appl. Probab.* 23.2, pp. 529–583 (cit. on pp. 7, 79, 81, 83, 85, 86, 96).
- Kang, H.-W., W. R. KhudaBukhsh, H. Koepl, and G. A. Rempała (2018). "Quasi-steady-state approximations derived from a stochastic enzyme kinetics model". In: Submitted (cit. on p. 10).
- Karatzas, I. and S. E. Shreve (1991). *Brownian motion and stochastic calculus*. Second. Vol. 113. Graduate Texts in Mathematics. Springer-Verlag, New York (cit. on p. 101).
- Katakakis, M. N. and L. C. Smit (2012). "A SUCCESSIVE LUMPING PROCEDURE FOR A CLASS OF MARKOV CHAINS". In: *Probability in the Engineering and Informational Sciences* 26.4, pp. 483–508 (cit. on p. 136).
- Kemeny, J. G., J. L. Snell, et al. (1960). *Finite markov chains*. Vol. 356. van Nostrand Princeton, NJ (cit. on pp. 17, 18, 141).
- Kephart, J. O. and S. R. White (May 1993). "Measuring and modeling computer virus prevalence". In: *Research in Security and Privacy, 1993. Proceedings., 1993 IEEE Computer Society Symposium on*, pp. 2–15 (cit. on p. 116).
- Kesidis, G., Y. Shan, B. Ugaonkar, and J. Liebeherr (Sept. 2015). "Network Calculus for Parallel Processing". In: *SIGMETRICS Perform. Eval. Rev.* 43.2, pp. 48–50 (cit. on p. 13).
- KhudaBukhsh, W. R., A. Audy, Y. Disser, and H. Koepl (2018). "Approximate lumpability for Markovian agent-based models using local symmetries". In: Submitted (cit. on p. 10).
- KhudaBukhsh, W. R., S. Kar, A. Rizk, and H. Koepl (2018). "Provisioning and performance evaluation of parallel systems with output synchronization". In: Submitted (cit. on p. 10).
- KhudaBukhsh, W. R., J. Rueckert, J. Wulfheide, D. Hausheer, and H. Koepl (2017). "SCHEDMIX: Heterogeneous Strategy Assignment in Swarming-based Live Streaming". In: Submitted (cit. on p. 10).
- KhudaBukhsh, W. R., B. Alt, S. Kar, A. Rizk, and H. Koepl (July 2017). *Collaborative Uploading in Heterogeneous Networks: Optimal and Adaptive Strategies*. Extended version <http://arxiv.org/abs/1712.04175>. URL: <http://arxiv.org/abs/1712.04175> (cit. on p. 59).
- KhudaBukhsh, W. R., B. Alt, S. Kar, A. Rizk, and H. Koepl (Apr. 2018). "Collaborative Uploading in Heterogeneous Networks: Optimal and Adaptive Strategies". In: *IEEE International Conference on Computer Communications (INFOCOM)* (cit. on pp. 4, 10).
- KhudaBukhsh, W. R., A. Rizk, A. Frömmgen, and H. Koepl (2017). "Optimizing Stochastic Scheduling in Fork-Join Queueing Models: Bounds and Applications". In: *IEEE International Conference on Computer Communications (INFOCOM)* (cit. on pp. 10, 52).
- KhudaBukhsh, W. R., J. Rückert, J. Wulfheide, D. Hausheer, and H. Koepl (May 2016). "Analysing and leveraging client heterogeneity in swarming-based live streaming". In: *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, pp. 386–394 (cit. on pp. 10, 121, 122).
- KhudaBukhsh, W. R., J. Rückert, J. Wulfheide, D. Hausheer, and H. Koepl (Dec. 2015). *A Comprehensive Analysis of Swarming-based Live Streaming to Leverage Client Heterogeneity*. Tech. rep. Technische Universitaet Darmstadt, Germany (cit. on pp. 147–150).

- KhudaBukhsh, W. R., C. Woroszylo, G. A. Rempała, and H. Koepl (2018). “**Functional Central Limit Theorem For Susceptible-Infected Process On Configuration Model Graphs**”. In: Submitted (cit. on p. 10).
- Kim, H. and E. Gelenbe (2012). “Stochastic gene expression modeling with hill function for switch-like gene responses”. In: *IEEE/ACM Trans. Comput. Biol. Bioinform.* 9.4, pp. 973–979 (cit. on p. 97).
- Kim, J. K., K. Josić, and M. R. Bennett (2015). “The relationship between stochastic and deterministic quasi-steady state approximations”. In: *BMC Syst. Biol.* 9.1, p. 87 (cit. on p. 97).
- Kim, J. K., G. A. Rempała, and H.-W. Kang (2017). “Reduction for stochastic biochemical reaction networks with multiscale conservations”. In: *arXiv preprint arXiv:1704.05628* (cit. on pp. 79, 97).
- Kim, J. H., B. Sudakov, and V. H. Vu (2002). “**On the asymmetry of random regular graphs and random graphs**”. In: *Random Structures & Algorithms* 21.3-4, pp. 216–224 (cit. on pp. 119, 125).
- Kiss, I. Z., J. C. Miller, and P. L. Simon (2017). *Mathematics of Epidemics on Networks: From Exact to Approximate Models*. Vol. 46. Springer (cit. on p. 136).
- Kratochvíl, J., A. Proskurowski, and J. A. Telle (1998). “Complexity of graph covering problems”. In: *Nordic Journal of Computing* 5, pp. 173–195 (cit. on p. 136).
- Kuntz, J., P. Thomas, G.-B. Stan, and M. Barahona (2017). “Rigorous bounds on the stationary distributions of the chemical master equation via mathematical programming”. In: *arXiv preprint arXiv:1702.05468* (cit. on p. 137).
- Kurtz, T. G. (1972). “The relationship between stochastic and deterministic models for chemical reactions”. In: *J. Chem. Phys.* 57.7, pp. 2976–2978 (cit. on p. 87).
- Kurtz, T. G. (1981). *Approximation of Population Processes*. SIAM (cit. on p. 144).
- Laidler, K. J. (1955). “Theory of the transient phase in kinetics, with special reference to enzyme systems”. In: *Can. J. Chem.* 33.10, pp. 1614–1624 (cit. on p. 80).
- Lelarge, M. (2012). “**Diffusion and cascading behavior in random networks**”. In: *Games Econom. Behav.* 75.2, pp. 752–775 (cit. on p. 116).
- Lelarge, M. and J. Bolot (2008). “**A Local Mean Field Analysis of Security Investments in Networks**”. In: *Proceedings of the 3rd International Workshop on Economics of Networked Systems*. ACM, pp. 25–30 (cit. on p. 142).
- Łuczak, T. (1988). “**The automorphism group of random graphs with a given number of edges**”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 104.3, pp. 441–449 (cit. on pp. 119, 125).
- Martin, J. B. and Y. M. Suhov (Aug. 1999). “**Fast Jackson networks**”. In: *Ann. Appl. Probab.* 9.3, pp. 854–870 (cit. on pp. 72, 74, 75, 185).
- Mather, W. H., J. Hasty, L. S. Tsimring, and R. J. Williams (Dec. 2011). “**Factorized time-dependent distributions for certain multiclass queueing networks and an application to enzymatic processing networks**”. In: *Queueing Systems* 69.3, pp. 313–328 (cit. on pp. 16, 17).
- McKay, B. D. and N. C. Wormald (Dec. 1984). “**Automorphisms of random graphs with specified vertices**”. In: *Combinatorica* 4.4, pp. 325–338 (cit. on pp. 119, 125).
- Mesa, M. A., A. Ramírez, A. Azevedo, C. Meenderinck, B. Juurlink, and M. Valero (2009). “Scalability of Macrobloc-level Parallelism for H.264 Decoding”. In: *ICPADS*, pp. 236–243 (cit. on pp. 6, 25, 29).

- Meyer, P. A. (1962). "A decomposition theorem for supermartingales". In: *Illinois J. Math.* 6, pp. 193–205 (cit. on p. 101).
- Michaelis, L. and M. L. Menten (1913). "Die kinetik der invertinwirkung". In: *Biochem. Z.* 49.333–369, p. 352 (cit. on p. 80).
- Miller, J. C. (2011). "A note on a paper by Erik Volz: SIR dynamics in random networks [MR2358436]". In: *J. Math. Biol.* 62.3, pp. 349–358 (cit. on p. 101).
- Miller, J. C., A. C. Slim, and E. M. Volz (2012). "Edge-based compartmental modelling for infectious disease spread". In: *Journal of The Royal Society Interface* 9.70, pp. 890–906 (cit. on p. 101).
- Molloy, M. and B. Reed (1995). "A critical point for random graphs with a given degree sequence". In: *Proceedings of the Sixth International Seminar on Random Graphs and Probabilistic Methods in Combinatorics and Computer Science, "Random Graphs '93"* (Poznań, 1993). Vol. 6. 2-3, pp. 161–179 (cit. on p. 113).
- Molloy, M. and B. Reed (1998). "The size of the giant component of a random graph with a given degree sequence". In: *Combin. Probab. Comput.* 7.3, pp. 295–305 (cit. on p. 114).
- Mousavi, M., H. Al-Shatri, W. R. KhudaBukhsh, H. Koepl, and A. Klein (Sept. 2017). "Cross-Layer QoE-Based Incentive Mechanism for Video Streaming in Multi-Hop Wireless Networks". In: *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pp. 1–7 (cit. on p. 10).
- Mukhopadhyay, A., A. Karthik, and R. R. Mazumdar (2016). "Randomized Assignment of Jobs to Servers in Heterogeneous Clusters of Shared Servers for Low Delay". In: *Stochastic Systems* 6.1, pp. 90–131 (cit. on pp. 63, 69, 71, 72, 75, 185).
- Newby, J. M. (2012). "Isolating intrinsic noise sources in a stochastic genetic switch". In: *Phys. Biol.* 9.2, p. 026002 (cit. on p. 78).
- Newby, J. M. (2015). "Bistable switching asymptotics for the self regulating gene". In: *J. Phys. A* 48.18, p. 185001 (cit. on p. 78).
- Ney, P. and E. Nummelin (Apr. 1987). "Markov Additive Processes II. Large Deviations". In: *The Annals of Probability* 15.2, pp. 593–609 (cit. on pp. 34, 38, 171, 173).
- Nijholt, E., B. Rink, and J. Sanders (2016). "Graph fibrations and symmetries of network dynamics". In: *Journal of Differential Equations* 261.9, pp. 4861–4896 (cit. on p. 128).
- Nisan, N., T. Roughgarden, É. Tardos, and V. V. Vazirani (2007). *Algorithmic Game Theory*. Cambridge University Press (cit. on p. 203).
- Nodelman, U., C. R. Shelton, and D. Koller (2002). "Continuous time Bayesian networks". In: *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 378–387 (cit. on p. 138).
- Norris, N. (1995). "Universal covers of graphs: Isomorphism to depth  $n-1$  implies isomorphism to all depths". In: *Discrete Applied Mathematics* 56.1, pp. 61–74 (cit. on p. 126).
- Pastor-Satorras, R. and A. Vespignani (2002). "Epidemic Dynamics in Finite Size Scale-free Networks". In: *Physical Review E* 65.3 (cit. on p. 142).
- Pedersen, M. G., A. M. Bersanib, and E. Bersanic (July 2006). "The Total Quasi-Steady-State Approximation for Fully Competitive Enzyme Reactions". In: *Bulletin of Mathematical Biology* 69.1, p. 433 (cit. on pp. 90, 188–190).
- Perez-Carrasco, R., P. Guerrero, J. Briscoe, and K. M. Page (2016). "Intrinsic noise profoundly alters the dynamics and steady state of morphogen-controlled bistable genetic switches". In: *PLoS Comput. Biol.* 12.10, pp. 1–23 (cit. on p. 78).

- Polato, I., R. Ré, A. Goldman, and F. Kon (2014). "A Comprehensive View of Hadoop Research-A Systematic Literature Review". In: *Journal of Network and Computer Applications* 46, pp. 1–25 (cit. on pp. 4, 11).
- Poloczek, F. and F. Ciucu (2014). "Scheduling Analysis with Martingales". In: *Performance Evaluation* 79, pp. 56–72 (cit. on pp. 23, 161).
- Poloczek, F. and F. Ciucu (2016). "Contrasting Effects of Replication in Parallel Systems: From Overload to Underload and Back". In: *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*. SIGMETRICS '16. ACM, pp. 375–376 (cit. on p. 50).
- Rand, D. A. (2009). "Correlation Equations and Pair Approximations for Spatial Ecologies". In: *Advanced Ecological Theory*. Blackwell Publishing Ltd., pp. 100–142 (cit. on p. 116).
- Rao, C. V. and A. P. Arkin (2003). "Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm". In: *J. Chem. Phys.* 118.11, pp. 4999–5010 (cit. on p. 97).
- Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning*. The MIT Press (cit. on p. 41).
- Rebolledo, R. (1980). "Central limit theorems for local martingales". In: *Z. Wahrsch. Verw. Gebiete* 51.3, pp. 269–286 (cit. on pp. 101, 104, 106, 108).
- Rejaie, R. and N. Magharei (2014). "On Performance Evaluation of Swarm-based Live Peer-to-Peer Streaming Applications". In: *Springer Multimedia Systems* 20.4 (cit. on p. 139).
- Richerzhagen, B., J. Wulfheide, H. Koeppl, A. Mauthe, K. Nahrstedt, and R. Steinmetz (June 2016). "Enabling crowdsourced live event coverage with adaptive collaborative upload strategies". In: *Proc. IEEE 17th Int. Symp. A World of Wireless, Mobile and Multimedia Networks*, pp. 1–3 (cit. on p. 2).
- Riordan, O. and N. Wormald (2010). "The diameter of sparse random graphs". In: *Combinatorics, Probability and Computing* 19.5–6, pp. 835–926 (cit. on p. 128).
- Rizk, A., F. Poloczek, and F. Ciucu (June 2015). "Computable Bounds in Fork-Join Queueing Systems". In: *SIGMETRICS Perform. Eval. Rev.* 43.1, pp. 335–346 (cit. on pp. 5, 12, 13, 25, 30, 44, 52).
- Rizk, A., F. Poloczek, and F. Ciucu (2016). "Stochastic bounds in Fork-Join queueing systems under full and partial mapping". In: *Queueing Systems* 83.3, pp. 261–291 (cit. on pp. 25, 44).
- Rubino, G. and B. Sericola (1989). "On weak lumpability in Markov chains". In: *Journal of Applied Probability* 26.3, pp. 446–457 (cit. on p. 17).
- Rubino, G. and B. Sericola (1993). "A finite characterization of weak lumpable Markov processes. Part II: The continuous time case". In: *Stochastic Processes and their Applications* 45.1, pp. 115–125 (cit. on pp. 17, 19).
- Rückert, J., B. Richerzhagen, E. Lidanski, R. Steinmetz, and D. Hausheer (2015). "TopT: Supporting Flash Crowd Events in Hybrid Overlay-based Live Streaming". In: *IFIP NETWORKING* (cit. on p. 139).
- Sanft, K. R., D. T. Gillespie, and L. R. Petzold (2011). "Legitimacy of the stochastic Michaelis–Menten approximation". In: *IET Syst. Biol.* 5.1, pp. 58–69 (cit. on p. 97).
- Sauro, H. M. and B. N. Kholodenko (2004). "Quantitative analysis of signaling networks". In: *Prog. Biophys. Mol. Biol.* 86.1, pp. 5–43 (cit. on p. 96).

- Schneider, K. R. and T. Wilhelm (2000). "Model reduction by extended quasi-steady-state approximation". In: *J. Math. Biol.* 40.5, pp. 443–450 (cit. on p. 79).
- Schnell, S. and P. K. Maini (2000). "Enzyme kinetics at high enzyme concentration". In: *Bull. Math. Biol.* 62.3, pp. 483–499 (cit. on pp. 81, 94).
- Schnell, S. and C. Mendoza (1997). "Closed form solution for time-dependent enzyme kinetics". In: *J. Theor. Biol.* 187.2, pp. 207–212 (cit. on p. 79).
- Segel, I. H. (1975). *Enzyme kinetics*. Vol. 360. Wiley, New York (cit. on pp. 77, 79).
- Segel, L. A. (1988). "On the validity of the steady state assumption of enzyme kinetics". In: *Bull. Math. Biol.* 50.6, pp. 579–593 (cit. on pp. 80, 86, 96, 187, 188).
- Segel, L. A. and M. Slemrod (1989). "The quasi-steady-state assumption: a case study in perturbation". In: *SIAM Rev.* 31.3, pp. 446–477 (cit. on pp. 79–82, 86).
- Sharma, N., D. K. Krishnappa, D. E. Irwin, M. Zink, and P. J. Shenoy (2013). "**Green-Cache: augmenting off-the-grid cellular towers with multimedia caches**". In: *Multimedia Systems Conference 2013, MMSys '13, Oslo, Norway, February 27 - March 01, 2013*, pp. 271–280 (cit. on p. 48).
- Shastri, S., A. Rizk, and D. Irwin (Nov. 2016). "**Transient Guarantees: Maximizing the Value of Idle Cloud Capacity**". In: *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 992–1002 (cit. on p. 48).
- Silva, P. da, J. Dias, and M. Ricardo (2016). "Mistrustful P2P: Privacy-preserving File Sharing Over Untrustworthy Peer-to-Peer Networks". In: *IFIP NETWORKING* (cit. on p. 153).
- Simon, P. L. and I. Z. Kiss (2012). "From exact stochastic to mean-field ODE models: a new approach to prove convergence results". In: *The IMA Journal of Applied Mathematics* 78.5, pp. 945–964 (cit. on p. 136).
- Simon, P. L., M. Taylor, and I. Z. Kiss (Apr. 2011). "**Exact epidemic models on graphs using graph-automorphism driven lumping**". In: *Journal of Mathematical Biology* 62.4, pp. 479–508 (cit. on pp. 9, 18, 119, 121, 122, 124, 136).
- Smith, S., C. Cianci, and R. Grima (2016). "Analytical approximations for spatial stochastic gene expression in single cells and tissues". In: *J. Royal Soc. Interface* 13.118 (cit. on p. 97).
- Šošić, A., W. R. KhudaBukhsh, A. M. Zoubir, and H. Koepl (May 2017a). "**Inverse Reinforcement Learning in Swarm Systems**". In: *AAMAS Workshop on Transfer in Reinforcement Learning* (cit. on p. 10).
- Šošić, A., W. R. KhudaBukhsh, A. M. Zoubir, and H. Koepl (May 2017b). "**Inverse Reinforcement Learning in Swarm Systems (Best Paper Award Finalist)**". In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (cit. on p. 10).
- Stewart, W. J. (2000). "Numerical methods for computing stationary distributions of finite irreducible Markov chains". In: *Computational Probability*. Springer, pp. 81–111 (cit. on p. 137).
- Stiefenhofer, M. (1998). "Quasi-steady-state approximation for chemical reaction networks". In: *J. Math. Biol.* 36.6, pp. 593–609 (cit. on p. 79).
- Subramanya, S., T. Guo, P. Sharma, D. E. Irwin, and P. J. Shenoy (2015). "SpotOn: A Batch Computing Service for the Spot Market". In: *SoCC*, pp. 329–341 (cit. on p. 23).
- Takahashi, Y. (1975). "A lumping method for numerical calculations of stationary distributions of Markov chains". In: *B-18, Department of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan* (cit. on p. 137).



- Thomasian, A. (Aug. 2014). "[Analysis of Fork/Join and Related Queueing Systems](#)". In: *ACM Comput. Surv.* 47.2, 17:1–17:71 (cit. on p. 5).
- Tian, T. and K. Burrage (2006). "Stochastic models for regulatory networks of the genetic toggle switch". In: *Proc. Natl. Acad. Sci. U.S.A.* 103.22, pp. 8372–8377 (cit. on p. 97).
- Twitter, Inc (2018). [Periscope](#). Accessed: July 1, 2017 (cit. on pp. 1, 53).
- Tzafriri, A. R. (2003). "Michaelis-Menten kinetics at high enzyme concentrations". In: *Bull. Math. Biol.* 65.6, pp. 1111–1129 (cit. on pp. 81, 88, 90).
- Van Slyke, D. D. and G. E. Cullen (1914). "The mode of action of urease and of enzymes in general". In: *J. Biol. Chem.* 19.2, pp. 141–180 (cit. on p. 81).
- Varadhan, S. R. S. (2016). *Large deviations*. American Mathematical Society (cit. on pp. 37, 38).
- Volz, E. (2008). "[SIR dynamics in random networks with heterogeneous connectivity](#)". In: *J. Math. Biol.* 56.3, pp. 293–310 (cit. on p. 101).
- Vulimiri, A., P. B. Godfrey, R. Mittal, J. Sherry, S. Ratnasamy, and S. Shenker (2013). "Low Latency via Redundancy". In: *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*. CoNEXT '13. New York, NY, USA: ACM, pp. 283–294 (cit. on pp. 50, 57).
- Wang, F., Y. Xiong, and J. Liu (2010). "mTreebone: A Collaborative Tree-Mesh Overlay Network for Multicast Video Streaming". In: *IEEE TPDS* 21.3 (cit. on p. 139).
- Watts, D. J. and S. H. Strogatz (1998). "Collective Dynamics of 'Small-world' Networks". In: *Nature* 393, pp. 440–442 (cit. on p. 151).
- Wen, Y. and J. Sun (Mar. 2007). "On Minimum-Delay Data Block Transport over Two-Connected Mesh Networks". In: *Proc. IEEE Wireless Commun. and Networking Conf.* Pp. 4034–4039 (cit. on p. 56).
- Wierman, J. C. and D. J. Marchette (2004). "[Modeling computer virus prevalence with a susceptible-infected-susceptible model with reintroduction](#)". In: *Comput. Statist. Data Anal.* 45.1, pp. 3–23 (cit. on p. 116).
- Yamashita, M. and T. Kameda (1996). "Computing on anonymous networks. I. Characterizing the solvable cases". In: *IEEE Transactions on Parallel and Distributed Systems* 7.1, pp. 69–89 (cit. on p. 126).
- Ying, L., R. Srikant, and S. Shakkottai (2010). "The Asymptotic Behavior of Minimum Buffer Size Requirements in Large P2P Streaming Networks". In: *IEEE Information Theory and Applications Workshop (ITA)* (cit. on pp. 146–148).
- Yoshihara, T., S. Kasahara, and Y. Takahashi (2001). "Practical Time-Scale Fitting of Self-Similar Traffic with Markov-Modulated Poisson Process". In: *Telecommunication Systems* 17.1, pp. 185–211 (cit. on pp. 33, 42, 45, 67).
- Zaharia, M., M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica (2010). "Spark: Cluster Computing with Working Sets". In: *USENIX HotCloud* (cit. on p. 4).
- Zhang, G., Y. Wen, J. Zhu, and Q. Chen (Nov. 2011). "On file delay minimization for content uploading to media cloud via collaborative wireless network". In: *Proc. Int. Conf. Wireless Commun. and Signal Process.* Pp. 1–6 (cit. on p. 56).
- Zhang, X. and H. Hassanein (2012). "A Survey of Peer-to-Peer Live Video Streaming Schemes - An Algorithmic Perspective". In: *Computer Networks* 56.15 (cit. on p. 139).
- Zhou, Y., D. M. Chiu, et al. (2007). "A Simple Model for Analyzing P2P Streaming Protocols". In: *IEEE ICNP* (cit. on pp. 146–148, 150, 151).

Zhou, Y., D.-M. Chiu, and J. Lui (2011). "A Simple Model for Chunk-scheduling Strategies in P2P Streaming". In: *IEEE/ACM TON* 19.1 (cit. on pp. [146–148](#), [150](#)).





## CURRICULUM VITÆ

---

# WASIUR RAHMAN KHUDA BUKHSH

### PERSONAL INFORMATION

DATE OF BIRTH	15 June, 1988
PLACE OF BIRTH	West Bengal, India
FAMILY STATUS	Single
HOME PAGE	<a href="https://wasiur.github.io/">https://wasiur.github.io/</a>

### EDUCATION

Master of Statistics (M.Stat.) Indian Statistical Institute, Kolkata, India	2009-2011
Bachelor of Science (B.Sc.) with Honours in Statistics University of Calcutta, Kolkata, India	2006-2009

### WORK EXPERIENCE

Technische Universität Darmstadt <i>Research Associate</i> , Bioinspired Communication Systems Darmstadt, Germany	2014-
ICICI Bank <i>Manager</i> , Advanced Analytics, Business Intelligence Unit (BIU) Mumbai, India	2011-2013

July 18, 2018



## ERKLÄRUNG LAUT §9 PROMOTIONSORDNUNG

---

### **§ 8 Abs. 1 lit. c PromO**

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

### **§ 8 Abs. 1 lit. d PromO**

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

### **§ 9 Abs. 1 PromO**

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

### **§ 9 Abs. 2 PromO**

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

---

Datum und Unterschrift